Analyse et commande des robots sous-marins autonomes à propulsion vectorielle reconfigurable

Loïck Degorre

COLLEGE DOCTORAL BRETAGNE — SCIENCES POUR L'INGENIEUR

Institut de Recherche Dupuy de Lôme UMR CNRS 6027

ÉCOLE NATIONALE D'INGÉNIEURS DE BREST

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE D'INGÉNIEURS DE BREST

ÉCOLE DOCTORALE N° 647
Sciences pour l'Ingénieur
Spécialité : *Robotique - Mécanique*

Par
Loïck DEGORRE

Analysis and control of autonomous underwater vehicles with re-configurable vectoring thrust.

Analyse et commande des robots sous-marins autonomes à propulsion vectorielle reconfigurable.

Thèse présentée et soutenue à Brest, le 11 octobre 2023
Unité de recherche : Institut de Recherche Dupuy de Lôme (IRDL) UMR 6027

Rapporteurs avant soutenance :

Prénom NOM	Fonction et établissement d'exercice
Prénom NOM	Fonction et établissement d'exercice
Prénom NOM	Fonction et établissement d'exercice

Composition du Jury :

Président :	Prénom NOM	Fonction et établissement d'exercice
Examinateurs :	Prénom NOM	Fonction et établissement d'exercice
	Prénom NOM	Fonction et établissement d'exercice
	Prénom NOM	Fonction et établissement d'exercice
	Prénom NOM	Fonction et établissement d'exercice
Dir. de thèse :	Emmanuel DELALEAU	Professeur des universités - ENIB
Encadrant :	Olivier CHOCRON	Maître de Conférence HDR - ENIB

Invité(s) :

Prénom NOM	Fonction et établissement d'exercice

RÉSUMÉ

La robotique marine a pris de plus en plus d'importance ces dernières années. Le nombre croissant d'applications en surface et sous l'eau incite les chercheurs et les ingénieurs à développer et à améliorer les technologies marines de pointe. La robotique marine est la clé des futures avancées en matière d'exploration marine, de surveillance de l'environnement, de gestion des ressources, d'énergie marine renouvelable et de bien d'autres applications encore à découvrir. À mesure que notre compréhension des océans s'approfondit, la demande en véhicules autonomes avancés capables d'opérer dans ces environnements hostiles n'a jamais été aussi forte. Des applications telles que l'inspection et la maintenance des éoliennes off-shore et des hydroliennes sous-marines sont l'un des maillons manquants de la chaîne qui faciliterait la démocratisation de ces technologies. La robotique marine est l'un des meilleurs candidats pour y parvenir.

Les deux principaux aspects de la robotique marine traités dans ce travail sont la manœuvrabilité et le contrôle. Ces deux aspects apparaissent comme des points clés pour débloquer de nouvelles applications. La question de la manœuvrabilité et de la mobilité est essentielle pour de nombreuses applications marines. La réponse habituelle pour augmenter la mobilité d'un véhicule est d'ajouter des moteurs.Mais, augmenter le nombre de moteurs augmente également le coût du véhicule et ajoute du poids, ce qui va à l'encontre de l'autonomie et de la facilité de manipulation et de déploiement. Pour concevoir des véhicules autonomes bon marché, économes en énergie et légers, le nombre de propulseurs doit être réduit. De nombreuses solutions existent en dehors du contexte maritime pour augmenter le nombre de degrés de liberté actionnés d'un véhicule sans ajouter de nouveaux actionneurs lourds. Pour les véhicules à roues, le *differential drive* et les roues *omni-directionnelles* permettent de libérer un véhicule des contraintes cinématiques des roues habituelles et débloquer de nouveaux degrés de mobilité.

Propulsion vectorielle reconfigurable

Pour les avions et les bateaux, la solution la plus courante pour améliorer la manœuvrabilité d'un véhicule est la propulsion vectorielle ou reconfigurable. L'idée de la propulsion reconfigurable est d'ajouter un ou plusieurs actionneurs légers et des dispositifs mécaniques pour permettre de diriger le vecteur de poussée produit par le propulseur ou le moteur principal. Ce type de technologie est notamment utilisé sur les avions à réaction, dont les tuyères vectorielles augmentent considérablement les capacités. Certains systèmes de poussée vectorielle permettent même des manœuvres contre-intuitives comme le décollage et l'atterrissage verticaux des avions à réaction.

La propulsion vectorielle est utilisée depuis de nombreuses années sur les véhicules marins. Le moteur hors-bord orientable d'un bateau particulier est un bon exemple de système de propulsion vectorielle. En tournant la roue, le pilote dirige le vecteur de poussée de l'hélice par l'intermédiaire d'un système mécanique, généralement à commande hydraulique. Un moteur principal et un actionneur supplémentaire (le pilote dans cet exemple) permettent d'actionner deux des trois degrés de liberté du bateau. Le principal avantage de ce type de technologie par rapport aux surfaces de contrôle traditionnelles est que le vecteur de poussée (amplitude et direction) est indépendant de la vitesse du véhicule. Cela permet une plus grande manœuvrabilité que les navires équipés de gouvernails, par exemple.

De nombreuses technologies de propulsion reconfigurable pour les véhicules marins autonomes suivent ce même principe. Mais, comme le montre ce travail, ces solutions ne sont pas exemptes de défauts, surtout lorsqu'elles sont mises en place sur de petits systèmes. Le principal défaut à l'utilisation de propulseurs vectoriel posittionnés à l'arrière du véhicule est l'apparition d'effets de couplage. Notamment, dans la majorité des cas, ces propulseurs sont utilisés pour générer des forces transversales excentrées créant des moments sur le corps du robot. Le plus souvent, les moments ainsi générés sont utilisés comme variable d'entrée dans le contrôle des véhicules mais les effets de forces transversaux sont négligés. La robustesse naturelle du véhicule aux perturbations de force en embardée et en pilonnement est alors mise en jeu. Sur des gros véhicules dont la coque est optimisée pour favoriser les mouvements d'avance et de rotation, les effets de force latérale sont principalement amortis. En revanche, sur de plus petits véhicules ayant des formes moins optimales (cylindrique ou sphérique par exemple), la stabilité naturelle en embardée et en pilonnement peut ne pas être assez robuste pour

encaisser et amortir les effets des forces perturbatrices. Dans ce cas, les propulseurs reconfigurables conventionnels ne sont pas indiqués car bien que la stabilité du système puissent être démontrée mathématiquement, ils sont susceptibles de créer eds comportements oscillants ou perturbés. Des essais en simulations montrent que ces effets de force perturbateurs peuvent empêcher le véhicule de remplir sa mission.

La preuve de concept d'une nouvelle solution est donc proposée dans ce travail et mise en pratique sur un nouveau robot : *PlaSMAR*. L'idée est d'avoir des paires de propulseurs parallèles se déplaçant et tournant autour de la coque du véhicule. L'effet combiné des deux propulseurs est contrôlé pour générer soit une force orientable et un moment stationnaire, soit pour diriger l'axe d'un moment avec une force constante. Étant donné que les propulseurs sont manipulés par paires parallèles, aucun effet de couplage n'est créé.

Le robot PlaSMAR est présenté comme une preuve de concept de cette nouvelle solution propulsive. L'étude de la matrice de configuration de propulsion (TCM) correspondant à deux configurations possible de PlaSMAR et les résultats de simulation exposés dans ce travail montrent que l'anneau rotatif permet effectivement d'ajouter un nouveau DOF au véhicule sans créer de nouveau effets de couplage perturbateur. Ce nouveau concept de reconfiguration semble donc être une solution mieux adaptée aux petits véhicules.

Les deux configurations de PlaSMAR présentées apportent de nouvelles solutions aux problèmes traditionnels de contrôle des véhicules sous-marins. La première configuration ajoute un nouveau degré de liberté actionné et indépendant des autres : l'embardée. Elle facilite les tâches de suivi de position.

La deuxième configuration permet d'actionner le tangage. Dans ce cas, ce nouveau degré de liberté peut être utilisé soit en tant que tel, pour générer un moment de tangage sur le véhicule, soit pour compenser l'angle de roulis du véhicule et maintenir les propulseurs montés sur l'anneau dans le plan horizontal du repère inertiel. Ces deux solutions améliorent les performances du véhicule pour les tâches présentées ici. Il faut aussi noter que l'anneau rotatif arrière avec deux propulseurs permet de reproduire l'actionnement en cavalement, tangage et lacet utilisé sur de nombreux véhicules sous-marins habituels.

Cette nouvelle technologie fera l'objet d'études plus approfondies. En particulier, l'actionnement de l'anneau rotatif n'a pas encore été réglé. De même, l'opération d'allocation des poussées n'est pas triviale sur ce système. Une première solution est néanmoins

présentée facilitant le contrôle. Cette solution ne prend pas en compte la dynamique de l'anneau ni celle des propulseurs. Des travaux supplémentaires sur l'hélice et la dynamique de reconfiguration sont nécessaires pour optimiser la répartition. L'un des axes à étudier est l'extension du modèle du système pour introduire le modèle de l'anneau. Ainsi, l'entrée du système ne sera plus uniquement le vecteur d'efforts appliqués au robot mais un vecteur contenant les poussées des propulseurs et l'angle du ou des anneaux rotatifs.

Commande

Le deuxième levier pour augmenter la maniabilité des systèmes robotiques marins est l'association de la commande et du guidage. De manière générale, la commande fait référence à l'ensemble des lois et des calculs utilisés pour calculer les entrées d'un système pour qu'il suive une tâche donnée. Sur les véhicules marins, les entrées sont généralement les forces générées par les propulseurs et les angles des éventuelles surfaces de contrôle ou des systèmes de propulsion vectorielle. Des calculs basés sur le modèle cinématique et dynamique du véhicule sont alors effectués et optimisés pour accomplir au mieux la tâche. La plupart des contrôleurs utilisés dans ce travail suivent l'approche basée modèle et supposent que l'état du véhicule et son modèle sont parfaitement mesurés. Il s'agit de deux hypothèses fortes mais nécessaires. L'estimation du modèle d'un véhicule marin et la mesure de son état sous l'eau sont deux sujets de recherche en cours. Pour que ces hypothèses soient raisonnables, les contrôleurs sont conçus pour être aussi robustes que possible vis-à-vis des approximations du modèle, des incertitudes, des perturbations et des erreurs de mesure.

Les contrôleurs sont associés à des principes de guidage transformant la tâche en références que le véhicule peut suivre. Hérités de la tradition navale, les algorithmes de guidage permettent par exemple de transformer une coordonnée géographiques en références d'avance et de cap pour la vaisseau. Ces références sont ensuite utilisées dans le contrôleur et le système est asservi dessus.

La solution la plus répandue pour le guidage est le guidage par *Ligne de vue* (LOS). La ligne de vue reproduit le comportement d'un capitaine expérimenté. Le principe est de pointer le véhicule vers une cible éloignée, un point de passage, et d'avancer tout droit dans sa direction. Ce comportement a été étendu au scénario de suivi de chemin où, au lieu de pointer directement vers le prochain point de passage, le véhicule est asservi sur

un chemin reliant deux points consécutifs. Il est donc pointé vers un point se déplaçant sur le chemin et donc l'évolution peut faire partie des variables du contrôle. Suivant l'héritage naval, la LOS est notamment adaptée au contrôle de véhicules actionnés en cavale en lacet en surface ou en cavalement, en tangage et en lacet sous l'eau. C'est la méthode la plus utilisée dans la recherche et l'industrie. Elle couvre facilement la majorité des tâches de suivi de chemin à grande échelle mais elle est moins adaptée aux tâche de suivi de trajectoire ou aux systèmes ayant des configurations d'actionnement différentes. Les références générées par le guidage en LOS sont ensuite fournies à un contrôleur, le plus souvent basé sur un PID. Les modèles des véhicules marins étant non-linéaires, les correcteurs PID sont souvent associés à des mécanismes de linéarisation. L'intérêt de cette procédure est d'annuler les termes non-linéaires du modèle dans la boucle fermée pour faciliter l'action du PID. Plusieurs méthodes existent utilisant soit les valeurs courantes des termes non-linéaires du modèle, soit des valeurs nominales désirées soit encore une association de termes courants et nominaux. Dans tous les cas, les termes non-linéaires de la boucle fermée sont annulé au voisinage de la trajectoire. Quand le véhicule s'écarte trop de la trajectoire en revanche les termes d'anticipations peuvent être déstabilisant. Dans les cas sous-actionnés où la LOS ne peut pas être utilisée, des mécanismes de guidage doivent être ajoutés. Deux solutions sont mises en avant dans ce travail. La première consiste à réduire l'espace de travail de manière à créer de nouvelles relations non-diagonales entre les degrés de libertés sollicités dans la tâche et les degrés de liberté du robot. La seconde, détaillée ci-après, consiste à introduire une matrice de guidage cinématique non-diagonale notée $\mathcal{H}$ qui permet de contrôler certains degrés de liberté non-actionnés au travers des couplages cinématiques. Le troisième type de contrôleur le plus commun en robotique marine est le contrôleur par *mode glissant* (SMC). L'avantage du SMC dans le cas des véhicules sous-actionnés est qu'il permet facilement d'intégrer le guidage au contrôle. En effet, par construction du contrôleur, il est possible de calculer les surface de glissement utilisées pour calculer l'entrée sur un degré de liberté à partir des signaux d'erreur d'un autre degré de liberté. Ce faisant, la construction même du contrôleur intègre des relations non-diagonales qui permettent de contrôler un degré de liberté non actionné au travers des couplages dynamiques du systèmes.

La première solution proposée introduit un nouveau principe de guidage cinématique basé sur le modèle. Le principe de cette méthode est relativement simple. Un point traquant est introduit sur le véhicule. Ce point peut représenter le point focal

d'une caméra ou le point d'action d'un effecteur. C'est ce point qui sera asservi sur la trajectoire à la place du centre du repère mobile traditionnellement utilisé. Ce point introduit de nouvelles relations cinématiques dans le modèles. En particulier, les vitesses linéaires en ce point sont en fait des fonctions de vitesses linéaires et angulaires du centre du véhicule par effet de levier. Étant donné que la plupart des véhicules marins sont actionnés en rotation par simplicité mais sont utilisés sur des tâches où le contrôle de la position est primordial, ces nouvelles relations sont particulièrement intéressantes. L'idée est donc de calculer des références de vitesses angulaires au centre du repère mobile du véhicule permettant de générer les vitesses linéaires demandées par la tâche au point traquant. Pour ce faire, une matrice de gain non diagonale basée sur le modèle cinématique du véhicule est introduite : $\mathcal{H}$.

L'avantage majeur de cette méthode est que le comportement du point traquant dans le repère inertiel peut être choisi exactement bien que les translations du véhicules ne soient pas actionnées directement. Le contrôleur peut donc être utilisé pour permettre à des véhicules sous-actionnés par rapport à une tâche de suivi de position de la remplir. Il est notamment appliqué sur des scénarios de suivi de trajectoire et de suivi de chemin. En suivi de trajectoire, la tâche est définie en terme de position, de vitesse et d'accélération et est contrainte temporellement. En suivi de chemin, le véhicule suit une route entre des points de passage sans contrainte temporel. Le premier scénario permet des applications à petite échelle mais de grande précision alors que le deuxième est plutôt adapté à des applications de grande échelle où la composante temporelle et le comportement du véhicule entre les points de passage ne sont pas prioritaires.

Les résultats de simulation montre que le contrôleur donne de très bons résultats sur les tâches auxquelles il est soumis. Notamment, il permet de contrôler le robot *RSM* actionné en cavalement, pilonnement, roulis et lacet sur des tâches sollicitant les trois translations et le cap. Le contrôleur permet d'exploiter le roulis du véhicule qui n'est pas requis dans la tâche pour compenser par le guidage l'absence d'actionnement en embardée. Il fourni en fait plusieurs solutions, une ou le roulis est utilisé pour la compensation et une ou c'est le lacet qui est utilisé. Bien que semblant plus naturelle, cette deuxième solution montre de moins bonnes performances sur les tâches testées. En effet, le degré de liberté utilisé pour la compensation ne peut pas être contrôlé. Le lacet étant principalement utilisé pour contrôlé le cap du véhicule dans le repère inertiel, il est perdu s'il est utilisé pour la compensation et la tâche ne peut pas être complètement remplie.

Le deuxième cas d'application intéressant de ce contrôleur est le contrpole du robot *Remus100* sur une application de suivi de chemin. Le robot Remus100 est représentatif de la principale classe de robots sous-marins: les robots torpilles actionnés en cavalement, tangage et lacet. Ces véhicules sont traditionnellement utilisés sur des tâches de grande échelle où ils suivent des chemins définis en position. Le contrôleur basé sur la matrice $\mathcal{H}$ a donc été appliqué au robot Remus100 sur l'une de ces tâches. Les résultats de simulation montre que malgré les limitations de rayons de braquage et de variations de vitesse de ce genre de véhicule, la tâche est parfaitement remplie. En fait, ce cas permet de montrer que ce nouveau principe de guidage permet de choisir facilement une solution pour le problème de suivi de particule. Ce problème est représentatif à la fois d'une problématique de suivi de chemin et de problématiques de type *leader/follower*. Le contrôleur pourrait par exemple être utilisé pour asservir un robot (follower) sur un autre système remplissant une tâche quelconque (follower). La démonstration de stabilité basée sur le critère de Lyapunov et obtenue par *back-stepping* est donnée dans ce travail.

Une deuxième solution mêlant guidage et contrôle est aussi présentée. Cette solution s'appuie sur le concept de la *platitude différentielle*. La platitude est une caractéristique inhérente d'un système qui permet d'établir une loi de contrôle basée modèle qui a été démontrée comme étant particulièrement robuste aux approximations de modèle et aux bruits de mesures. La platitude s'apparente à la notion de commandabilité d'un système linéaire mais pour les systèmes non-linéaire. La platitude semble donc tout indiquée pour le contrôle des véhicules marins et sous-marins.

Pour construire un contrôlé basé sur la platitude, il faut d'abord montrer que le système à contrôler est *plat*. Un système plat est un système dont l'ensemble des variables (état et entrées) peut être exprimé comme un ensemble de fonctions d'une sortie particulière nommé la *sortie plate* (et de ces dérivés) sans n'avoir à exprimer d'équation différentielle. Les équations de la platitude représentant les variables du système en fonction de la sortie plate sont utilisées pour établir un contrôleur basé modèle du système. Elles permettent aussi de calculer des valeurs désirées ou nominales des variables du systèmes utilisées dans le contrôleur pour la linéarisation.

Plusieurs exemples de la littérature montrent que le modèle représentant un vaisseau de surface n'est pas naturellement plat. Les termes de couplage de masses ajoutées et de frottement introduisent des couplages dynamiques dans le modèle qui empêchent de montrer la platitude. En revanche, un cas simplifié du vaisseau de surface, l'Hovercraft,

a une forme circulaire et une répartition de masse homogène qui permettent de montrer la platitude. L'idée est donc d'établir un contrôleur basé sur la platitude en utilisant le modèle de l'hovercraft puis de l'appliquer au modèle générique du vaisseau de surface en utilisant les propriétés naturelles de robustesse de la platitude.

Les résultats de simulation présentés dans ce travail montrent que le contrôleur calculé en passant par le modèle simplifié de l'hovercraft se comporte aussi bien sur l'hovercraft que sur le vaisseau de surface. Les performances de suivi de trajectoire dans le cas sans courant sont presque parfaites tant que le véhicules reste à proximité de la trajectoire. En revanche, et c'est une des faiblesse majeure du contrôle par platitude et par linéarisation par feedforward en général, quand le système s'écarte trop de la trajectoire désirée, la stabilité n'est plus assurée. En effet, comme les valeurs nominales des variables du modèle sont utilisées dans le contrôleur, les termes non-linéaires du systèmes dans la boucle fermée s'annule quand le système est au voisinage de la trajectoire mais deviennent de plus en plus important quand il s'en écarte. Aussi, bien que la platitude soit particulièrement robuste aux approximations de modèle, quand le véhicule simulé est très éloigné des hypothèses de forme et de distribution de masse de l'hovercraft (s'il est particulièrement plus long que marge par exemple) et qu'il est soumis à des perturbations comme un courant marin, le contrôleur atteint sa limite et peut diverger. Pour corriger ces deux écueils, la contrôle basé sur la platitude est associé à un contrôleur proportionnel dérivé *intelligent* (iPD).

Le iPD est un contrôleur adaptatif basé sur le PID. Il contient une fonction d'adaptation, en général basé sur les filtres dit *ALIEN*, qui estime à chaque échantillon les perturbations auxquelles le système est soumis sans les différencier. Tous les effets perturbateurs sont regroupés en une seule valeur ou un seul vecteur. Le contrôleur est calculé en utilisant un modèle linéaire dit *ultra-local* représentant le système pendant un très court instant. Le iPD est mis en avant comme étant une solution générale pour remplacer les correcteurs basés PID .

Ici, le iPD est calculé sur le système représentant la différence entre la commande du système et la commande nominale calculée par la platitude. Il compense les comportements problématiques du contrôleur basé sur la platitude quand le système est trop loin de la trajectoire ou que les différences entre le modèle réel et le modèle de l'hovercraft sont trop grandes. Les performances du correcteurs associant platitude et iPD sont démontrées en simulation. L'ajout du iPD permet effectivement d'assurer la stabilité du système même quand il s'éloigne de la trajectoire. Le iPD permet aussi d'estimer et

d'annuler les effets non modélisés dus aux différences de modèle et aux perturbations extérieures augmentant ainsi la robustesse du contrôleur de départ.

Jusqu'à présent, le contrôleur basé sur la platitude avec le iPD n'a été appliqué qu'aux vaisseaux de surface. Des travaux supplémentaires permettront assez facilement de l'appliquer au modèle du véhicule sous-marin. Considérant les résultats prometteurs sur le vaisseau de surface, le contrôleur semble adapté à la commande des véhicules sous-marins actionnés en cavalement, tangage et lacet pour des tâches de suivi de trajectoires en position.

Conclusion

Ce travail s'inscrit à la suite d'une longue série de travaux sur la conception et le contrôle des véhicules marins. De nombreux types de lois de contrôle et de principes de guidage pour les véhicules de surface et sous-marins ont été introduits et étudiés en détail. Toutes les solutions issues de la littérature montrent des performances théoriques équivalentes. Elles garantissent toutes la stabilité globale ou semi-globale du système, même en présence de perturbations externes et de mesures bruitées.

Des travaux supplémentaires seront nécessaires pour comparer concrètement les différentes méthodes, à la fois sur le plan mathématique et sur des véhicules expérimentaux. Il est très difficile de classer les méthodes car leurs performances dépendront probablement du système lui-même. D'autres paramètres devront être pris en compte, comme le temps de calcul, les effets de l'échantillonnage numérique ou l'impact de mesures de mauvaise qualité.

Un nouveau principe de guidage cinématique basé sur la matrice pratique $\mathcal{H}$ est introduit. Ce nouveau principe permet de contrôler un véhicule sous-actionné ou mal actionné en utilisant les couplages cinématiques entre une translation non actionnée et une rotation actionnée. La commande est asymptotiquement stable et robuste aux perturbations externes et aux bruits de mesure. Elle est facilement étendue à d'autres types de systèmes grâce aux règles de conception et à l'algorithme de calcul de la matrice fournis. Ces résultats permettent également d'évaluer les capacités d'un AUV donné et de déterminer les tâches pour lesquelles il est adapté.

Le contrôleur a été appliqué avec succès dans des scénarios de suivi de trajectoire et de suivi de chemin. Le suivi de trajectoire a été effectué sur le Remus100, un AUV représentant la classe des navires uqr en forme de torpille. Le contrôleur basé sur la

matrice $\mathcal{H}$ permet de résoudre le problème de suivi de trajectoire de plusieurs manières. L'utilisation du contrôleur basé sur la matrice $\mathcal{H}$ permet de choisir avec précision le comportement du point de suivi du véhicule. Dans le cas du suivi de trajectoire, par exemple, il permet de trouver des solutions simples basées sur le PID pour le critère de stabilité de Lyapunov. Dans de futurs travaux, le contrôleur pourra être utilisé dans d'autres applications du back-stepping pour accomplir différents types de tâches.

L'application du contrôleur à un véhicule physique serait utile pour évaluer ses performances en conditions réelles. Il est notamment très difficile d'anticiper le comportement de la loi de contrôle en ce qui concerne l'échantillonnage numérique, la latence du calculateur embarqué ou des erreurs de mesure irrégulières.

Un contrôleur basé sur la platitude a également été développé dans le cas simple des navires de surface. Le contrôle basé sur la platitude semble être un concurrent de taille pour le contrôle des véhicules marins car il s'avère très robuste à l'approximation du modèle et aux perturbations externes. Dans ce travail, un contrôleur basé sur la platitude a été calculé pour les navires de surface en utilisant le modèle simplifié de l'hovercraft. Le contrôleur a été développé dans l'hypothèse d'une forme de coque circulaire et d'une distribution uniforme de la masse, mais il a été appliqué avec succès à un navire de surface générique.

Le contrôleur basé sur la platitude comprend naturellement un principe de guidage. Il permet de calculer une force de poussée et un moment de lacet à partir de deux signaux de position. Les équations de platitude obtenues dans cet exemple pourraient également être utilisées comme principe de guidage avec un autre type de contrôleur.

Pour accroître la robustesse et contrecarrer le comportement potentiellement problématique de la commande basée sur la platitude quand le système s'éloigne de la trajectoire souhaitée, le contrôleur a été associée à un correcteur proportionnel dérivé intelligent. L'iPD augmente la robustesse déjà impressionnante du contrôleur basé sur la platitude et résout efficacement les problèmes causés par la linéarisation par anticipation lorsque le véhicule se trouve en dehors de la limite acceptable autour de la trajectoire souhaitée. Le contrôleur combiné est stable et résistant aux perturbations externes et aux bruit de mesure.

Enfin, de nouveaux résultats sur la poussée vectorielle appliquée aux véhicules légers ont été présentés. Des simulations ont notamment été menées pour montrer que l'utilisation de propulseurs vectoriels excentrés n'est pas adaptée à tous les types de véhicules. Bien qu'ils soient naturellement bornés, les effets de force de couplage générés par les

propulseurs vectoriels peuvent compromettre la mission des véhicules trop légers ou des véhicules qui n'ont pas de résistance naturelle aux perturbations en force transversale.

Un nouveau mode de propulsion vectorielle pour les véhicules sous-marins autonomes a également été présenté. Alors que les propulseurs vectoriels sont généralement utilisés pour contrôler les moments par le biais de couplages dynamiques, les résultats de simulation montrent qu'ils peuvent également être utilisés pour contrôler les forces. Ce nouveau mode de contrôle en force s'avère particulièrement utile pour les missions où le contrôle en moment n'est naturellement pas adapté. C'est notamment le cas des missions autonomes d'amarrage ou de vol stationnaire

La preuve de concept d'un nouveau type de système de poussée vectorielle a également été proposée. Le nouveau système repose sur des paires de propulseurs parallèles montés sur un anneau capable de tourner autour de sections cylindriques de la coque d'un véhicule. Ce mouvement de rotation peut être utilisé pour diriger le vecteur de force généré par la paire de propulseurs tout en maintenant un axe de moment stationnaire. Lorsque les propulseurs sont alignés sur l'axe de rotation de l'anneau, il est possible de diriger l'axe du moment généré par la paire de propulseurs tout en maintenant le vecteur de force aligné sur le corps du véhicule. Il est démontré que ce deuxième mode débloque aussi un nouveau DOF pour le véhicule et permet de compenser les perturbations dues au roulis.

La prochaine étape consiste à concevoir et à construire une deuxième version de la plateforme d'essai et à étudier les solutions possibles pour l'actionnement de l'anneau. Des couplages magnétiques optimisés seront étudiés car ils permettraient de piloter la rotation de l'anneau depuis l'intérieur de la coque, ce qui éviterait de percer des trous dans la coque.

D'autres travaux seront menés sur le contrôleur de ce véhicule. Un contrôleur prenant en compte la dynamique de l'anneau serait intéressant et pourrait s'avérer nécessaire. Le modèle du véhicule devrait être étendu pour prendre en compte le comportement de l'anneau et inclure l'angle de l'anneau comme nouvelle entrée de commande. De même, des recherches supplémentaires sur l'optimisation de la procédure d'attribution sont nécessaires pour améliorer les performances.

Dans l'ensemble, ce travail propose plusieurs solutions pour accroître la manœuvrabilité des embarcations marines sous-actionnées grâce à la commande et à de nouvelles conceptions mécaniques.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PUBLICATIONS AND COMMUNICATIONS

Published International Journal papers

[IJ–1] Loïck **Degorre**, Emmanuel Delaleau, and Olivier Chocron. A survey on model-based control and guidance principles for autonomous marine vehicles. *Journal of Marine Science and Engineering*, 11(2), 2023. soumis 29/12/22, révisé 30/01/23, accepté le 9/02/23, publié 16/02/23.

International communications

[IC–1] Loïck **Degorre**, Olivier Chocron, and Emmanuel Delaleau. A new general approach for model-based control of underactuated auv based on kinematic coupling. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, October 23–27 2022. IEEE. (Poster).

Communications

[C–1] Loïck **Degorre**. Enhancing mobility of underactuated AUV through control and reconfigurable propulsion. In *ENIB International Mini Symposium on Robotics*, Brest, 11 May 2023.

[C–2] Loïck **Degorre**. Plateforme autonome sous-marine reconfigurable. In *Submeeting 2023*, Guerlédan, 27 April 2023.

[C–3] Loïck **Degorre**, Olivier Chocron, and Emmanuel Delaleau. Une approche générique de la commande basée modèle des AUV sous actionnés. In *25^{e} Congrès Français de Mécanique*, Nantes, 20 august to 2 september 2022.

[C–4] Loïck **Degorre**. Commande non linéaire basée modèle des robots sous-marins sous-actionnés. In *IRDL Seminar*, Brest, 02 June 2022.

Submitted International Journal papers

[SIJ–1] Loïck **Degorre**, Thor I. Fossen, Emmanuel Delaleau, and Olivier Chocron. A model-based kinematic guidance method for control of underactuated autonomous underwater vehicles. *Ocean Engineering*, 2023. (Under Review).

Popular science events

[PSE–1] Loïck **Degorre**. La robotique sous-marine. In *Fête de la Science*, Brest, 7 to 10 October 2021.

INTRODUCTION

Marine robotics, has gained increasing significance in recent years. The rapidly growing number of surface and underwater applications boosts both researchers and engineers to develop and enhance cutting-edge marine technologies. Marine robotics is the key to breakthroughs marine exploration, environmental monitoring, resource management, marine renewable energy and so many more applications yet to discover. As our understanding of the world's oceans deepens, the demand for advanced autonomous vehicles capable of operating in these harsh and remote environments has never been greater.

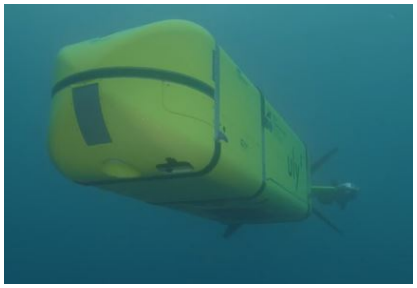
Since the early years of the cold war, military researchers all over the world saw in marine robotics an opportunity to extend their control over the seas. Cold war years saw some of the first autonomous marine systems both on the surface and underwater. Inheritance from these early works still shows to this day in the technologies, shapes, approaches and methods used in marine robotics.

Nowadays, marine robotics also appears as an essential tool for green energy production, observation and protection of marine environments. Applications like inspection and maintenance of off-shore wind turbines and underwater tidal turbines are one of the missing chain link that would facilitate the democratization of these technologies. Marine robotics is one of the best candidates to fulfill them.

The recent effort to improve our knowledge of the marine environments also raises a new class of applications for marine robotics. Cutting-edge autonomous and remotely operated vehicles are at work all around the globe to observe, measure and sometimes reconstruct the underwater world. Majors actors of research and industry keep developing new marine vehicle. This is notably the case of *Ifremer* [1] with the *Ulyx* shown on figure 1a and designed for medium range underwater mapping applications. Another example is the *Auto-Hover 1* which recently joined the largest fleet of autonomous vehicles in Europe at the National Oceanography Center in UK and shown on figure 1b.

The main two aspects of marine robotics treated in this work are maneuverability and control. Both are believed to be key points to unlock new applications. The matter

1. Institut Français de recherche pour l'exploitation de la mer, France



(a) Ulyx (Credit : IFREMER/ECA)

(b) Auto-Hover 1 (Credit: NOC)

Figure 1 – Examples of recent AUVs

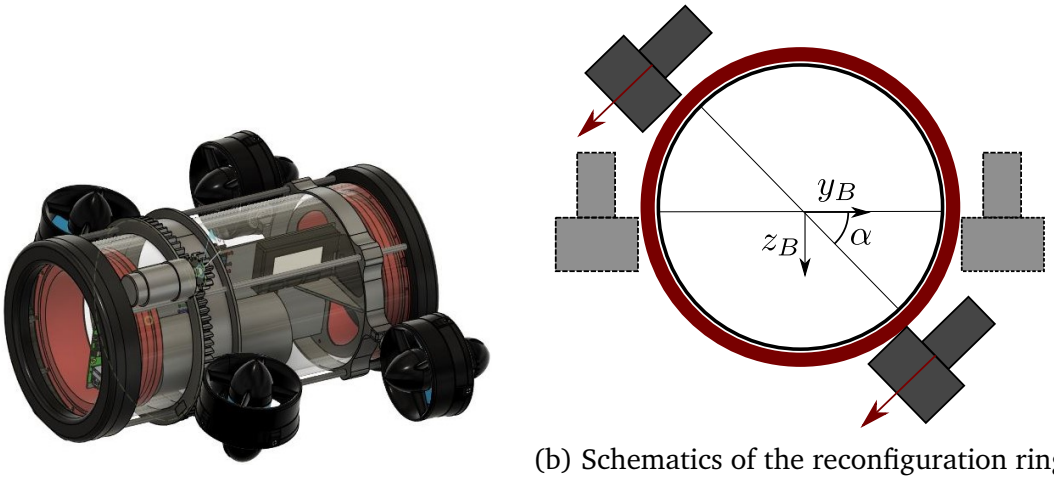
of maneuverability and mobility is essential for many marine applications. The usual answer to increasing the mobility of a vehicle is to add more motors. Increasing the number of thrusters also increases the cost of the vehicle and adds weight which goes against autonomy and ease of handling and deployment. In order to design cheap, energy-efficient and lightweight autonomous vehicles, the number of thrusters must be kept low. Many solutions exist outside of the marine context to increase the number of actuated degrees of freedom of a vehicle without adding new heavy-weight actuators. For wheeled vehicles, differential drive and *omni-directional* or *mecanum* wheels appear as a good solution to free a vehicle from the kinematic constraints of usual wheels and unlock new degrees of mobility as on the *Uranus* robot [124] displayed on figure 2. Mecanum wheels are notably used in warehouse applications where fast, precise and mobile vehicles are needed.

For planes and boats, the most common solution to increase the maneuverability of a vehicle is *thrust vectoring*. The idea of vector thrust is to add one or several lightweight actuators and mechanical contraptions to allow directing the thrust vector produced by the main thruster or engine. This kind of technology is notably used on jet planes where vectoring nozzles drastically increase their capabilities. Some vector thrust systems even allow vertical take off and landing of jet planes like the one shown on figure 3.

Thrust vectoring has been used for years on marine vehicles. The off-board steerable engine of a common boat is a good example of vector thrust system. When turning the wheel, the pilot directs the thrust vector of the propeller through a mechanical system, usually hydraulically driven. One main engine and a lightweight actuator (the pilot in this example) allow actuating two of the three degrees of freedom of the boat. The main advantage of this type of technology over the traditional control surfaces is that the thrust vector (magnitude and direction) is independent from the speed of the

Figure 2 – The Uranus omni-directional mobile robot.

Figure 3 – A vertical take-off and landing nozzle for jet planes.



(a) 3-D render of the PlaSMAR robot.

(b) Schematics of the reconfiguration ring principle.

Figure 4 – Proof of concept of the new vectoring thrust system : the PlaSMAR robot.

vehicle, allowing much greater maneuverability than ships with rudders for instance.

Many thrust vectoring technologies have been developed over the years for surface and underwater autonomous vessels [43]. These solutions opened new applications for small and lightweight vehicles. Some of these solutions are displayed and used in this work. Most of them consists in having one main thruster or propeller directed using one or two additional lightweight actuators, effectively increasing the number of actuated degrees of freedom of a single thruster to two or three. But, as shown in this work, these solutions are not without flaws and especially when working on small vehicles. The proof of concept to a new solution is then exposed in this work. This new solution consists in having pairs of parallel thrusters moving around the hull of the vehicle. The combined effect of the two thrusters is controlled to generate either a steerable force and a stationary moment or to direct the axis of a moment with a steady force. A 3-D render of the vehicle designed with technology can be seen on figure 4 along with schematics of the principle. The new solutions is applied in simulation for trajectory tracking and shown to effectively increase the capabilities of the vehicle on common tasks.

While this work mainly focuses on conventional propulsive technologies with thrusters with propellers, more and more alternatives propulsive technologies appear. Notably, bio-mimetic vehicles taking the shape of fish or turtles can be found in the literature and show promising mobility capacities [18, 180, 143]. The design and control of this new class of vehicles represent a whole new challenge.

The second lever to increase the maneuverability of marine robotic systems is *control*. Broadly, control refers to the ensemble of laws and calculations used to calculate the forces and moments required for a system to follow a given task. On marine craft, the control inputs are usually the forces generated by the thrusters and the angles of possible control surfaces or vectoring thrust systems. Calculations based on the kinematic and dynamic model of the vehicle are then performed and optimized to perform the task at best. Most of the controllers used in this work follow the model-based approach and assume that the state of the vehicle and its model are thoroughly measured. These are two strong but necessary hypotheses. Estimating the model of a marine vehicle and measuring its state underwater are two ongoing research subjects. To make these hypotheses sensible, the controllers are designed to be as robust as possible w.r.t. model approximations, uncertainties, disturbances and measurement errors. In marine robotics, and especially with poorly actuated vehicles, controllers are associated with guidance principles transforming the task into references that the vehicle can track.

After an in-depth study of the solutions proposed in the literature, two new control methods are proposed. These two new solutions blend guidance and control together. The first solution is a model-based kinematic guidance principle and dynamic controller based on a non-diagonal gain matrix denoted $\mathcal{H}$. This controller is used in simulation on underactuated vehicles to perform path following and trajectory tracking tasks. The second controller is based on differential flatness. This controller has in facts been developed on a simple surface vessel, the hovercraft system, but is shown to be applicable to most surface vehicles.

Overall, the controllers designed in this work are shown to increase the capabilities of underactuated systems. They unlock new tasks and behaviors for these vehicles and are believed to be a significant step ahead for marine robotics.

This manuscript is organized as follows. First, the model used to described the vehicles is explained in details along some remarks on the simulator developed for this work. Then, a bibliographical review of existing guidance methods and control principles is proposed. In the third chapter, the new kinematic guidance principle is described along several simulation results showing its capacities. Then, the flatness-based controller is detailed. Finally, the new vectoring thrust principle is presented with some additional results on reconfigurable thrust and illustrated through simulations of the PlaSMAR AUV.

CHAPTER 1

MODELING AND TASK DESCRIPTION

Contents

This chapter introduces the model used in this work to describe the behavior of a marine craft in its environment. The modeling of marine craft has received a lot of attention over more than a century. While the first studies on the motion of solid bodies through fluids can be found in ancient Greece, the work of Sir Horace Lamb in 1879 [101] is often cited as one of the first modern studies on the topic. The cold war years saw a great amount of work produced on the modeling of submarines in the U.S. and

notably the work of Imlay in the 60s [89]. About thirsty years later, the modeling of marine vehicles is popularized among the robotics community by the work of Fossen [63, 65, 64] which is the main reference used for modeling in the present work as in many other underwater robotics publications. The recent work of Antonelli [8] as also been a great inspiration for this work.

The goal of this chapter is to settle on the model and hypotheses used throughout this work for the modeling of the vehicles. It is not meant to give new results on the pure modeling. Nonetheless, some remarks on design and modeling choices are made in this chapter. It is very important to settle on a realistic and effective model for the system as it will later be used as a base for control algorithms in numerical simulations.

The model presented in this chapter is in six dimensions and represents both surface and underwater vehicles. In this work, surface vehicles are considered as a reduction of the underwater case to the horizontal plane, therefore neglecting some of the rotations happening outside of the plane (roll and pitch). As will be seen in the following examples, it is common practice to neglect some rotational effects of the model if the associated restoring torques are considered large enough to maintain the vehicle in an almost horizontal position or to neglect uncontrolled yet stable DOF.

This chapter also redefines the notion of task. It introduces a different understanding of the tasks based on the trajectory, constraints and environmental disturbances. This definition is used to define the requirements of a task in terms of actuated DOF or natural dynamic of the vehicle. It also introduces the different scenarios dictating the time constraints of the task.

1.1 Modeling of Marine Vehicles

1.1.1 Framework

Before introducing the model equations, it is mandatory to define the frames used in this work. The most common approach is to define two frames $\mathcal{R}_0$and $\mathcal{R}_B$:

— $\mathcal{R}_0(\boldsymbol{O}, \boldsymbol{x_o}, \boldsymbol{y_o}, \boldsymbol{z_o})$ is the inertial, North-East-Down (NED) frame

— $\mathcal{R}_B(\boldsymbol{O_B}, \boldsymbol{x_B}, \boldsymbol{y_B}, \boldsymbol{z_B})$ is a body-fixed frame

The origin of the body-fixed frame is usually chosen at the center of buoyancy of the vehicle $\boldsymbol{P_B}$ but can, in-facts, be chosen anywhere on the vehicle. The $\boldsymbol{x_B}$ axis of the mobile frame is most often aligned with the axis of greatest length of the vehicle.

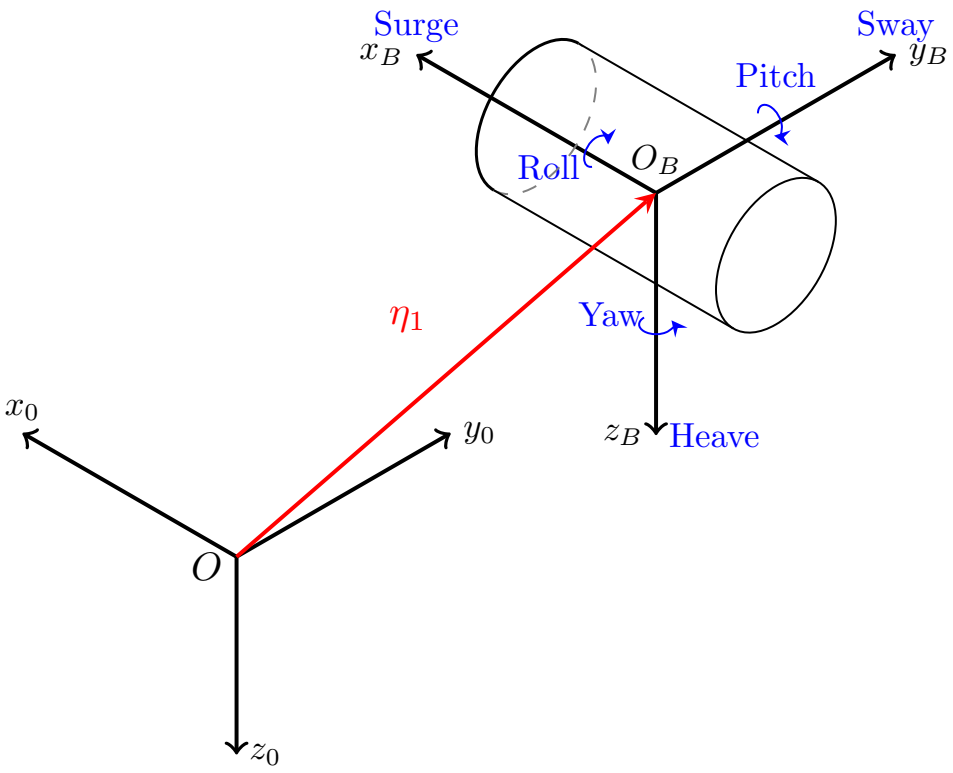


Figure 1.1 – Representation of the $\mathcal{R}_0$and $\mathcal{R}_B$frames.

A third frame $\mathcal{R}_p(\boldsymbol{O_p}, \boldsymbol{x_p}, \boldsymbol{y_p}, \boldsymbol{z_p})$ is introduced as it can be useful in some control applications: the *path-fixed* frame. This third frame will be redefined when used in the following examples of this work but it can be seen as a *Serret-Frenet* frame, centered on a virtual particle on the desired path, generally with $\boldsymbol{x_p}$ tangent to the path. The particle is associated with a propagation function disctating its behavior on the path. As will be seen later in this work, the propagation function can be parameterized and optimized in control applications.

The position of the vehicle in $\mathcal{R}_0$is denoted by the vector $\boldsymbol{\eta}_1 = [x\ y\ z]^\top$ and the orientation of the vehicle is given in the Roll-Pitch-Yaw convention and denoted $\boldsymbol{\eta}_2 = [\phi\ \theta\ \psi]^\top$. The position and orientation of the vehicle are regrouped in the vector $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^\top\ \boldsymbol{\eta}_2^\top]$.

Remark 1.1 *The Euler (ZYX) angle convention used to represent the orientation of the body-fixed frame is known to be singular in $\theta = \pm\pi/2$. This singularity notably appears in the definition of the transformation matrix $\boldsymbol{J}(\boldsymbol{\eta})$ (See section 1.1.2). To avoid the singular point, a quaternion representation can be used and is given in [63, 8]. In this work, the*

Translations along axis			**Rotations around axis**		
Axis	**Notation**	**Name**	**Axis**	**Notation**	**Name**
$\boldsymbol{x_B}$	u	Surge	$\boldsymbol{x_B}$	p	Roll
$\boldsymbol{y_B}$	v	Sway	$\boldsymbol{y_B}$	q	Pitch
$\boldsymbol{z_B}$	w	Heave	$\boldsymbol{z_B}$	r	Yaw

Table 1.1 – Usual names and notations for the motion of the marine craft (SNAME 1950)

Euler angle representation will be kept as much as possible as it is believed to be a more natural representation of the orientation of the vehicle. This is a sensible choice considering that most vehicles are naturally stabilized in the horizontal plane, only few applications in the literature require the vehicle to go perfectly vertical and that none of the scenarios described in this work do so. For applications operating close to the singular attitude, a quaternion approach is advised and some examples presented later in this work use it.

1.1.2 Kinematic Model

The velocity of the vehicle with respect to the inertial frame and expressed in the body-fixed frame is given by the vector $\boldsymbol{\nu} = [\boldsymbol{\nu}_1^\top \ \boldsymbol{\nu}_2^\top]$ where $\boldsymbol{\nu}_1 = [u \ v \ w]^\top$ is the vector of linear velocity of point O_B w.r.t. the earth-fixed frame and expressed in $\mathcal{R}_B$ and $\boldsymbol{\nu}_2 = [p \ q \ r]^\top$ is the vector of angular velocity of the mobile frame $\mathcal{R}_B$. The standard notation for the motions of the vehicle are given in Table 1.1, the SNAME [1] notation for marine vessels will be used as much as possible in this work.

The kinematic model of the vehicle is given by the velocity transformation:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{1.1}$$

1. Society of Naval Architects and Marine Engineers

where $\boldsymbol{J}(\boldsymbol{\eta})$ is the transformation matrix defined as:

$$\boldsymbol{J}(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{J}_1(\boldsymbol{\eta}) & 0 \\ 0 & \boldsymbol{J}_2(\boldsymbol{\eta}) \end{bmatrix} \tag{1.2a}$$

$$\boldsymbol{J}_1(\boldsymbol{\eta}) = \boldsymbol{R}(\boldsymbol{x_o}, \phi)\boldsymbol{R}(\boldsymbol{y_o}, \theta)\boldsymbol{R}(\boldsymbol{z_o}, \psi) \tag{1.2b}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\phi\sin\theta\sin\psi & -\cos\psi\sin\phi + \sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}$$

$$\boldsymbol{J}_2(\boldsymbol{\eta}) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \tag{1.2c}$$

In equation (1.1), $\boldsymbol{R}(i, j)$ denotes the rotation matrix of angle j around axis i. More details about the calculations used to establish $\boldsymbol{J}_2(\boldsymbol{\eta})$ can be found in [63]. As remarked before, equation (1.2c) shows a singularity in $\theta = \pm\pi/2$ because of the Euler angle representation. Quaternion alternative to $\boldsymbol{J}(\boldsymbol{\eta})$ can be found in [63, 8].

1.1.2.1 Addition of a tracking point

Some of the control methods used in this work require the definition of an intermediary point called a *tracking point*. The idea of having a tracking point is to control a point of the vehicle, different from the center of the moving frame, towards the desired state. This solution is often used when working with external manipulators or sensors and this tracking point can be the focal point of a camera or the grasping point of the end-effector of a manipulator arm. Thus, a tracking point E of coordinates $[\varepsilon_x \ \varepsilon_y \ \varepsilon_z]^\top$ in $\mathcal{R}_B$is introduced. Figure 1.2 displays an example of a tracking point E placed at the

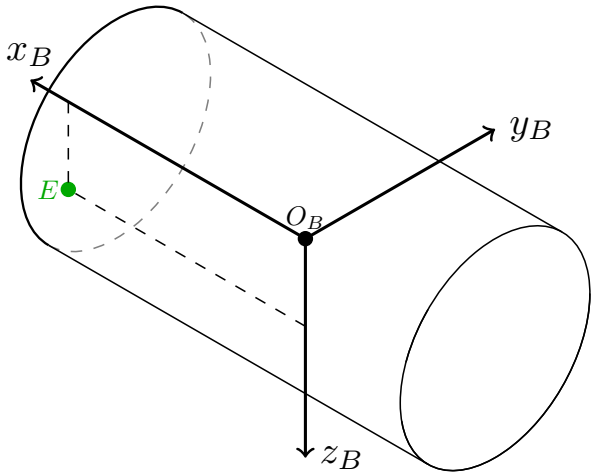


Figure 1.2 – Example of a tracking point E at the nose of the body of an underwater vehicle.

nose of the vehicle slightly below the horizontal plane. In this example the coordinates of E are $[\varepsilon_x\ 0\ \varepsilon_z]^\top$ in $\mathcal{R}_B$.

The kinematic model equation (1.1) is modified to include this new tracking point :

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{T}\boldsymbol{\nu} \tag{1.3}$$

where $\boldsymbol{T}$ is the transformation matrix moving the vector of linear and angular velocities from point O_B to point E:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & \varepsilon_z & -\varepsilon_y \\ 0 & 1 & 0 & -\varepsilon_z & 0 & \varepsilon_x \\ 0 & 0 & 1 & \varepsilon_y & -\varepsilon_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.4}$$

Matrix $\boldsymbol{T}$ is the matrix form equivalent to the Varignon transform applied to the kinematic wrench [167].

Note that the introduction of the transformation matrix $\boldsymbol{T}$ creates a new set of kinematic couplings in the model that will be used later in this work for the control of the vehicle.

1.1.3 Dynamic Model

The conventional dynamic model of marine craft is issued from Newton's second law and is usually written as:

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{1.5}$$

In equation (1.5), $\boldsymbol{\tau}(t) \in \mathbb{R}^6$ is the vector of propulsive forces and moments generated by the thrusters and applied to the vehicle, $\boldsymbol{M} \in \mathbb{R}^{6,6}$ is the mass and inertia matrix, $\boldsymbol{C}(\boldsymbol{\nu}) \in \mathbb{R}^{6,6}$ is the matrix of Coriolis, centripetal and gyroscopic effects, $\boldsymbol{D}(\boldsymbol{\nu}) \in \mathbb{R}^{6,6}$ is the matrix of damping effects and $\boldsymbol{g}(\boldsymbol{\eta}) \in \mathbb{R}^6$ is the vector of gravitational and buoyancy effects. The model matrices are given in detail in the following sections. Numerical values of the model parameters for some AUVs used in this work are given in appendix D.

The vector $\boldsymbol{\tau}$ expresses the forces and moments generated by the thrusters and control surfaces of the vehicle, it is written as:

$$\boldsymbol{\tau} = \begin{bmatrix} F_X \\ F_Y \\ F_Z \\ T_X \\ T_Y \\ T_Z \end{bmatrix} \tag{1.6}$$

1.1.3.1 Added masses

In order to take some of the environmental effects into account in the dynamic model and notably the hydrodynamic effects of the surrounding water, it is common practice in the robotics community to use the method of *added masses* [63][100][98]. Added masses are a set of mass coefficients, linearly added to the mass of the solid body

moving through water to simulate some of its effects. They are pressure-induced forces and moments due to the body impelling water movement. The coefficients appear in the mass matrix $\boldsymbol{M}$ and in the Coriolis and Centripetal matrix $\boldsymbol{C}(\boldsymbol{\nu})$. A lot of work has been put into calculating, measuring or estimating added masses over the years and the interested reader is referred to the following works for a deeper understanding of the concept of added masses [89, 88, 100] as well as some of the work done on the estimation of these coefficients [121, 98, 72].

The superposition of the body mass effects and the surrounding water effects is done linearly in the $\boldsymbol{M}$ and $\boldsymbol{C}(\boldsymbol{\nu})$ matrices calculations as:

$$\boldsymbol{M} = \boldsymbol{M}_b + \boldsymbol{M}_a \tag{1.7a}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_b(\boldsymbol{\nu}) + \boldsymbol{C}_a(\boldsymbol{\nu}) \tag{1.7b}$$

where $\boldsymbol{M}_b$ and $\boldsymbol{C}_b(\boldsymbol{\nu})$ are the body effects while $\boldsymbol{M}_a$ and $\boldsymbol{C}_a(\boldsymbol{\nu})$ are the added mass effects. The rigid-body terms $\boldsymbol{M}_b$ and $\boldsymbol{C}_b(\boldsymbol{\nu})$ are, for any vehicle, of the shape:

$$\boldsymbol{M}_b = \begin{bmatrix} m\mathbb{I}_{3,3} & -m\boldsymbol{S}(\boldsymbol{P_G}) \\ m\boldsymbol{S}(\boldsymbol{P_G}) & \boldsymbol{I_o} \end{bmatrix} \tag{1.8a}$$

$$\boldsymbol{C}_b(\boldsymbol{\nu}) = \begin{bmatrix} 0_{3,3} & -m\boldsymbol{S}(\boldsymbol{\nu}_1) - m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{P_G}) \\ -m\boldsymbol{S}(\boldsymbol{\nu}_1) - m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{P_G}) & m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{\nu}_1)\boldsymbol{P_G}) - \boldsymbol{S}(\boldsymbol{I_o}\boldsymbol{\nu}_2) \end{bmatrix} \tag{1.8b}$$

In equation (1.8), $\mathbb{I}_{3,3}$ is the 3 by 3 identity matrix, m is the dry mass of the body, $\boldsymbol{I_o}$ is the matrix of inertia of the body of the vehicle defined in $\mathcal{R}_B$and $\boldsymbol{P_G}$ is the position of the center of gravity in $\mathcal{R}_B$. The inertia matrix of the cylinder is:

$$\boldsymbol{I_o} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} + m\boldsymbol{S}(\boldsymbol{P_G})^\top \boldsymbol{S}(\boldsymbol{P_G}) \tag{1.9}$$

I_x, I_y, and I_z are the principal moments of inertia of the vehicle. They can be calculated using standard formulations.

Matrix $\boldsymbol{S}()$ is the skew-symmetric matrix defined as:

$$\boldsymbol{S}(\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^\top) = \begin{bmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{bmatrix} \tag{1.10}$$

The matrix $\boldsymbol{S}()$ represents the cross product such that $S(X)Y = X \prod Y$. The added mass effects depend on the shape of the hull of the vehicle. The vehicles used in this work are, for most of them, approximated with cylinders and hence have three planes of symmetry. The added mass matrices are then of shape:

$$\boldsymbol{M}_a = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & -Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & -Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -N_{\dot{r}} \end{bmatrix} \tag{1.11a}$$

$$\boldsymbol{C}_a(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \tag{1.11b}$$

The cylindrical shape approximation gives reliable simulation results and is widely spread in the literature. This hypothesis is supported by the fact that off-diagonal terms of $\boldsymbol{M}_a$ are generally of lesser magnitude than the diagonal terms, they are often negli-

gible. The off-diagonal added masses are also particularly hard to measure estimate or calculate.

For surface vessels, the shapes of the model matrices are different. Off-diagonal terms accounting for the coupled effects of sway and yaw due to the usual hull shape of surface vessels are added in the added mass matrix. In the horizontal plane space reduction and considering the mobile frame centered on the center of gravity of the surface vessel, the mass and added mass matrices are [63]:

$$\boldsymbol{M}_b = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{1.12a}$$

$$\boldsymbol{M}_a = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y_{\dot{r}} & -N_{\dot{r}} \end{bmatrix} \tag{1.12b}$$

The non-zero off-diagonal terms of $\boldsymbol{M}_a$ imply a different formulation of the added lass effects in the Coriolis and Centripetal matrix:

$$\boldsymbol{C}_a(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & X_{\dot{u}}u & 0 \end{bmatrix} \tag{1.13}$$

1.1.3.2 Hydrodynamic Damping

As for the added masses introduced above, hydrodynamic damping evaluation of marine craft is a difficult subject that has been much written about. The most common approximations used in the underwater robotics and control community are also used in this work. This is why, when talking about underwater vehicles of cylindrical hull, the damping effects will be considered fully decoupled and only linear and quadratic terms on the diagonal are used. Additional damping models are introduced and compared in

[74]. In this case, matrix $\boldsymbol{D}(\boldsymbol{\nu})$ is:

$$\begin{aligned}\boldsymbol{D}(\boldsymbol{\nu}) = &-\text{diag}(X_u, Y_v, Z_w, K_p, M_q, N_r) \\ &- \text{diag}(X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|, K_{p|p|}|p|, M_{q|q|}|q|, N_{r|r|}|r|)\end{aligned} \quad (1.14)$$

where X_u, ..., N_r are linear damping coefficients and $X_{u|u|}$, ..., $N_{r|r|}$ are quadratic coefficients.

For surface vehicles, a better rendition of the usual hull shape is given by the following damping matrix:

$$\boldsymbol{D}(\boldsymbol{\nu}) = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \quad (1.15)$$

This second damping representation allows taking the natural coupling between sway and yaw into account while maintaining an independent surge mode. The linear damping approximation is generally used with surface vessels but a quadratic damping term of the same shape could be added.

1.1.3.3 Gravitation and buoyancy effects

The vector $\boldsymbol{g}(\boldsymbol{\eta})$ contains the gravitation and buoyancy forces and moments undergone by the vehicle and expressed at the center of the vehicle in $\mathcal{R}_B$. These terms, referred to as *restoring torques and moments*, depend on the relative position of the centers of gravity $\boldsymbol{P_G}$ and buoyancy $\boldsymbol{P_B}$ in $\mathcal{R}_B$and on the orientation of the vehicle w.r.t. $\mathcal{R}_0$. Generally speaking, the center point of the moving frame $\boldsymbol{O_B}$ is chosen at the center of buoyancy $\boldsymbol{P_B}$ which often matches the geometrical center of the hull. On the other hand, the mass distribution of the vehicle would ideally be designed to have the center of gravity underneath the center of buoyancy. Point $\boldsymbol{P_G}$ would be of coordinates $(0, 0, z_G)$ in $\mathcal{R}_B$with z_G in the magnitude of a few centimeters to a meter depending on the vehicle. This disposition allows having stable roll and pitch modes for the vehicle.

The gravity and buoyancy forces $\boldsymbol{f_G}(\boldsymbol{\eta})$ and $\boldsymbol{f_B}(\boldsymbol{\eta})$ for a submerged vehicle are given

in the mobile frame as:

$$\boldsymbol{f_G}(\boldsymbol{\eta}) = \boldsymbol{J}_1(\boldsymbol{\eta})^{-1} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{1.16a}$$

$$\boldsymbol{f_B}(\boldsymbol{\eta}) = \boldsymbol{J}_1(\boldsymbol{\eta})^{-1} \begin{bmatrix} 0 \\ 0 \\ -\rho g V \end{bmatrix} \tag{1.16b}$$

with g the acceleration of gravity, ρ the local water density and V the volume of displaced water. Forces $\boldsymbol{f_G}(\boldsymbol{\eta})$ and $\boldsymbol{f_B}(\boldsymbol{\eta})$ are expressed at the centers of gravity and buoyancy respectively. The complete vector of gravity and buoyancy effects is then given at the center of the mobile frame as:

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{f_G}(\boldsymbol{\eta}) + \boldsymbol{f_B}(\boldsymbol{\eta}) \\ \boldsymbol{P_G} \times \boldsymbol{f_G}(\boldsymbol{\eta}) + \boldsymbol{P_B} \times \boldsymbol{f_B}(\boldsymbol{\eta}) \end{bmatrix} \tag{1.17}$$

Note that most of the vehicles used in this work are positively buoyant in sea water: $m < \rho V$.

1.2 Propulsive configuration

This section focuses on the representation of the propulsive arrangement of the vehicles used in this work. The two main actuator types used in this work are the fixed and the reconfigurable thruster. These two technologies are best suited for applications requiring high maneuverability. However, it is worth mentioning the great number of autonomous vehicles equipped with control surfaces (stern planes or rudders). Rudders, often associated with one or several fixed thrusters, are notably useful for long range applications with low turning radius where maneuverability is not a priority. Notably, underwater gliders used for long range oceanographic applications are generally equipped with control planes and active ballasts. Their mobility is limited but they are

very energy-efficient. Yet, because the action of control surfaces depends on the relative velocity of the craft with respect to the surrounding waters, this kind of actuation is not suited for applications like station keeping, precise maneuvers around a target or sharp rotations which are the main focus of this work.

1.2.1 Thruster Configuration Matrix

The *Thruster Configuration Matrix* (TCM) represents the thruster arrangement of the vehicle [67, 30]. It is used to calculate the combined propulsive effort of the n thrusters of the vehicle. The TCM, often denoted as $\boldsymbol{B}$, is also used for *thrust allocation*, it allows calculating the necessary thrust of each thruster u_i needed to generate a required effort vector. All the thrust forces are regrouped in the thrust vector $\boldsymbol{u}$.

Matrix $\boldsymbol{B}$ has 6 rows and one column per thruster. Each column of the TCM is built with the effort wrench of a thruster moved to the center of the vehicle. A local frame $\mathcal{R}_{\boldsymbol{P_i}}(\boldsymbol{P_i}, \boldsymbol{x}_{\boldsymbol{P_i}}, \boldsymbol{y}_{\boldsymbol{P_i}}, \boldsymbol{z}_{\boldsymbol{P_i}})$ is associated to thruster i. Frame $\mathcal{R}_{\boldsymbol{P_i}}$ is centered on the estimated thrust center of thruster i, $\boldsymbol{P_i}(x_{\boldsymbol{P_i}}, y_{\boldsymbol{P_i}}, z_{\boldsymbol{P_i}})$, and the axis $\boldsymbol{x}_{\boldsymbol{P_i}}$ is aligned with the axis of thrust.

The force generated by thruster i is expressed in $\mathcal{R}_{\boldsymbol{P_i}}$ as:

$$\boldsymbol{F}_{\boldsymbol{P_i}}^{\mathcal{R}_{\boldsymbol{P_i}}} = \begin{bmatrix} u_i & 0 & 0 \end{bmatrix} \tag{1.18}$$

where u_i is the i-th component of the vector containing all thrust norms of the n thrusters of the vehicle $\boldsymbol{u} \in \mathbb{R}^n$.

Frame $\mathcal{R}_{\boldsymbol{P_i}}$ is rotated with respect to $\mathcal{R}_B$ of an angle $\psi_{\boldsymbol{P_i}}$ around $\boldsymbol{z_B}$ and $\theta_{\boldsymbol{P_i}}$ around the newly created $\boldsymbol{y_B}'$ axis. The rotation matrix from $\mathcal{R}_{\boldsymbol{P_i}}$ to $\mathcal{R}_B$ is then:

$$R_{\boldsymbol{P_i}}^{B} = R(\psi_{\boldsymbol{P_i}}, \boldsymbol{z_B}) R(\theta_{\boldsymbol{P_i}}, \boldsymbol{y_B}') = \begin{bmatrix} \cos\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} & -\sin\psi_{\boldsymbol{P_i}} & \cos\psi_{\boldsymbol{P_i}} \sin\theta_{\boldsymbol{P_i}} \\ \sin\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} & \cos\psi_{\boldsymbol{P_i}} & \sin\psi_{\boldsymbol{P_i}} \sin\theta_{\boldsymbol{P_i}} \\ -\sin\theta_{\boldsymbol{P_i}} & 0 & \cos\theta_{\boldsymbol{P_i}} \end{bmatrix} \tag{1.19}$$

The force generated by thruster i expressed in $\mathcal{R}_B$ then becomes:

$$\boldsymbol{F}_{\boldsymbol{P_i}}^{\mathcal{R}_B} = R_{\boldsymbol{P_i}}^{B} \boldsymbol{F}_{\boldsymbol{P_i}}^{\mathcal{R}_{\boldsymbol{P_i}}} = \begin{bmatrix} \cos\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} \\ \sin\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} \\ -\sin\theta_{\boldsymbol{P_i}} \end{bmatrix} u_i \tag{1.20}$$

Finally, the moment created at the center of the vehicle by the force $\boldsymbol{F}_{\boldsymbol{P_i}}^{\mathcal{R}_B}$ is calculated at the center of the vehicle in $\mathcal{R}_B$ as:

$$T_{\boldsymbol{O_B}}^{\mathcal{R}_B} = \boldsymbol{P_i} \times \boldsymbol{F}_{\boldsymbol{P_i}}^{\mathcal{R}_B} = \begin{bmatrix} -\sin\theta_{\boldsymbol{P_i}} y_{\boldsymbol{P_i}} - \sin\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} z_{\boldsymbol{P_i}} \\ \cos\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} z_{\boldsymbol{P_i}} + \sin\theta_{\boldsymbol{P_i}} x_{\boldsymbol{P_i}} \\ \sin\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} x_{\boldsymbol{P_i}} - \cos\psi_{\boldsymbol{P_i}} \cos\theta_{\boldsymbol{P_i}} y_{\boldsymbol{P_i}} \end{bmatrix} u_i \tag{1.21}$$

Equation (1.20) and equation (1.21) give the i-th column of $\boldsymbol{B}$ and, building the rest of $\boldsymbol{B}$ iteratively with the other thrusters, the following relationship is established:

$$\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{u} \tag{1.22}$$

Finally, to perform the *thrust allocation*, the relation (1.22) must be "inverted". The following examples show that, even though the fully-actuated case is trivial with $\boldsymbol{B}$ being square and invertible, the underactuated case is not.

1.2.1.1 Thrust Allocation in the Fully actuated case

In the fully-actuated case, $n = 6$ and $\boldsymbol{B}$ invertible ($\det(\boldsymbol{B}) \neq 0$), the thrust allocation operation is easily realized inverting relation (1.22). It is then possible to know the required thrust force of each thruster to generate a given vector $\boldsymbol{\tau}$:

$$\boldsymbol{u} = \boldsymbol{B}^{-1}\boldsymbol{\tau} \tag{1.23}$$

An example of a fully-actuated vehicle is displayed on figure 1.3. This vehicle has six independent thrusters with no redundancies. The position and orientation of the six thrusters are given in table 1.2 where L is the length of the cynlindrical hull of the vehicle and R is the radius. This vehicle is theoretical, its thruster arrangement would

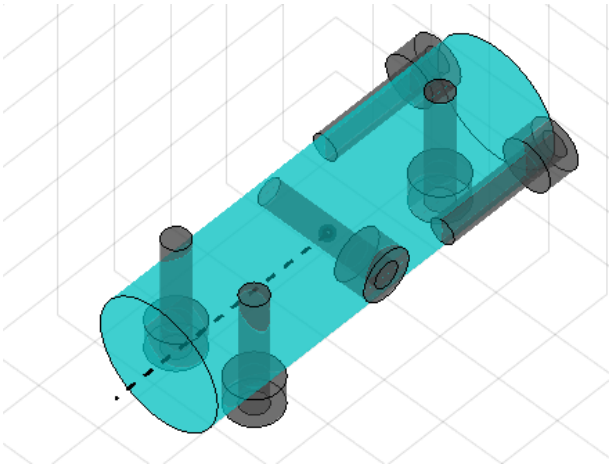


Figure 1.3 – Simulation render of the fully actuated example vehicle. Blue cylinder: Hull, black cylinders: Thrusters

Thruster 1	**Thruster 2**	**Thruster 3**	**Thruster 4**	**Thruster 5**	**Thruster 6**
$\begin{bmatrix} -L/3 \\ -R \\ 0 \end{bmatrix}$	$\begin{bmatrix} -L/3 \\ R \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} L/3 \\ -2R/3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} L/3 \\ 2R/3 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -L/3 \\ 0 \\ 0 \end{bmatrix}$
$\theta_{P_1} = 0$	$\theta_{P_2} = 0$	$\theta_{P_3} = 0$	$\theta_{P_4} = \pi/2$	$\theta_{P_5} = \pi/2$	$\theta_{P_6} = \pi/2$
$\psi_{P_1} = 0$	$\psi_{P_2} = 0$	$\psi_{P_3} = \pi/2$	$\psi_{P_4} = 0$	$\psi_{P_5} = 0$	$\psi_{P_6} = 0$

Table 1.2 – Propulsive configuration of the fully-actuated vehicle

be hard to physically carry out but it is a good example of the effects demonstrated here. Examples of fully-actuated vehicles can be seen on figure 1.

The TCM of this vehicle is:

$$\boldsymbol{B}_{FA} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2R/3 & -2R/3 & 0 \\ 0 & 0 & 0 & L/3 & L/3 & -L/3 \\ R/3 & -R/3 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.24}$$

All 6 DOF of the vehicle are actuated and can be controlled independently which is confirmed by the rank of the matrix: $\text{rank}(\boldsymbol{B}_{FA}) = 6$. The matrix $\boldsymbol{B}_{FA}$ is invertible: $\det(\boldsymbol{B}_{FA}) = -16R^2L/27$. We see that the first two thrusters both generate surge force and yaw moment but because they are coupled together, these two DOF are made independent. Same goes for the three last thrusters sharing heave, roll and pitch. The third thruster on the other hand, controls sway on its own and, because it is centered on the hull, does not generate any moment.

1.2.1.2 Thrust Allocation in the Underactuated case

In the underactuated case, $n < 6$, the thrust allocation is non-trivial. The TCM is not square hence not invertible. In order to describe the solutions for thrust allocation of underactuated vehicles, the *RSM* AUV is described in [169, 31, 30, 168]. The *RSM* AUV is equipped with four thrusters: $n = 4$. Two are placed at the rear of the vehicle and aligned with the $\boldsymbol{x_B}$ axis and two are placed in the middle of vehicle on each side and aligned with the $\boldsymbol{z_B}$ axis. The *RSM* vehicle is displayed on figure 1.4 and the position and orientation of theactuators are given in table 1.3.

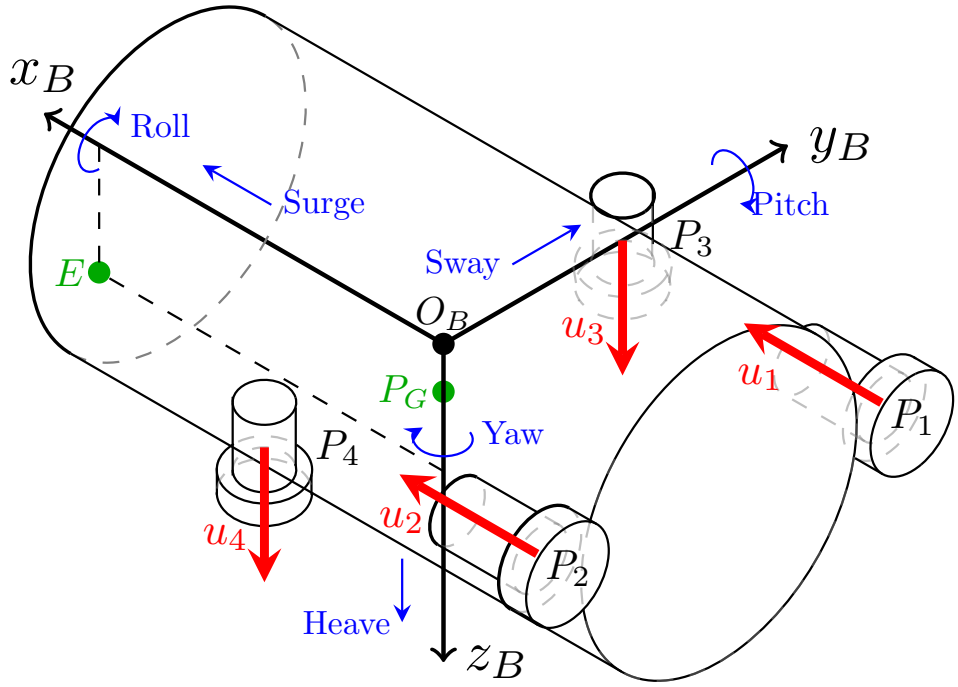


Figure 1.4 – Schematic of the *RSM* robot. Red vectors are unitary thrust vector of each thruster.

Thruster 1	**Thruster 2**	**Thruster 3**	**Thruster 4**
$\begin{bmatrix} -L/2 \\ R \\ 0 \end{bmatrix}$	$\begin{bmatrix} -L/2 \\ -R \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ R \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ -R \\ 0 \end{bmatrix}$
$\theta_{P_1} = 0$	$\theta_{P_2} = 0$	$\theta_{P_3} = -\pi/2$	$\theta_{P_4} = -\pi/2$
$\psi_{P_1} = 0$	$\psi_{P_2} = 0$	$\psi_{P_3} = 0$	$\psi_{P_4} = 0$

Table 1.3 – Propulsive configuration of the *RSM* robot

The TCM for robot *RSM* is then:

$$\boldsymbol{B}_{RSM} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & R & -R \\ 0 & 0 & 0 & 0 \\ -R & R & 0 & 0 \end{bmatrix} \tag{1.25}$$

It is clear in equation (1.25) that this vehicle is not able to generate any sway force (second row is zero) nor pitch moment (fifth row also zero). The vehicle is clearly underactuated and one gets: $\text{rank}(\boldsymbol{B}_{RSM}) = 4$. The first two thrusters share surge (addition of forces) and yaw (difference of forces). The last two share heave and roll in the same way.

The two most common solutions for thrust allocation in the underactuated case are the Moore-Penrose pseudo-inverse [129] and reduction of the calculation space. The principle of space reduction is to discard $6 - n$ lines of the TCM to make it square and invertible. The resulting reduced matrix must be of full rank showing that the vehicle is in facts fully-actuated on the reduced space. The discarded lines correspond to the non-actuated DOF of the vehicle or to the non-controlled DOF in the case of coupled DOF (see example of section 1.2.2). In this case, the lines 2 and 5 of $\boldsymbol{B}_{RSM}$ would be discarded. The reduced matrix becomes:

$$\boldsymbol{B}^{r}_{RSM} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -R & R \\ -R & R & 0 & 0 \end{bmatrix} \tag{1.26}$$

Matrix $\boldsymbol{B}^r_{RSM}$ is invertible $\det(\boldsymbol{B}^r_{RSM}) = 4R^2$ and its inverse is:

$$(\boldsymbol{B}^r_{RSM})^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & -1/R \\ 1 & 0 & 0 & 1/R \\ 0 & 1 & -1/R & 0 \\ 0 & 1 & 1/R & 0 \end{bmatrix} \tag{1.27}$$

Note that, when using the reduced TCM, the effort vector must also be reduced removing the same lines.

The second solution is to use the left-hand Moore-Penrose pseudo-inverse defined for a matrix A as [129]:

$$A^\dagger = (A^\top A)^{-1} A^\top \tag{1.28}$$

The pseudo-inverse of $\boldsymbol{B}_{RSM}$ is then:

$$(\boldsymbol{B}_{RSM})^\dagger = \frac{1}{2}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1/R \\ 1 & 0 & 0 & 0 & 0 & 1/R \\ 0 & 0 & 1 & -1/R & 0 & 0 \\ 0 & 0 & 1 & 1/R & 0 & 0 \end{bmatrix} \tag{1.29}$$

The Moore-Penrose pseudo-inverse proposes a least error solution for the inversion problem. Note that, because of the rows of zeros of $\boldsymbol{B}_{RSM}$ the space-reduction and pseudo-inverse give two very similar solutions. In this case, one gets:

$$\boldsymbol{u} = (\boldsymbol{B}_{RSM})^\dagger \boldsymbol{\tau} = (\boldsymbol{B}^r_{RSM})^{-1} \boldsymbol{\tau}^r \tag{1.30}$$

with $\boldsymbol{\tau}^r$ the reduced effort vector. Additional examples are displayed in section 5.1 where the Moore-Penrose pseudo-inverse is not satisfying and the reduced inverse must be used.

More advanced solutions based on quadratic programming could be used to deal with the inversion of the thrust allocation equation.

1.2.2 Reconfigurable thrusters

Reconfigurable thrusters are mechanical devices where a main propeller is coupled with additional mechanisms allowing reconfiguration of the thrust vector. They can also be referred to as *vector* or *vectorized* thrusters. An overview of different reconfiguration technologies can be found in [43]. This section is mainly focused on thrusters able to direct their thrust axis in one or two dimensions but this concept has been pushed forward in chapter 5 where thrusters able to rotate around the hull of the vehicle are studied.

The off-board engine of a typical private boat is a good everyday-life example of a 2D reconfigurable thruster. The propeller axis can be directed around the vertical axis allowing the engine to generate vector thrust in all the horizontal plane. Reconfigurable thrusters increase the number of actuated DOF of a vehicle with additional lightweight actuators but does not required additional heavyweight motors. It is a sensible solution to increase the mobility of a vehicle without reducing its autonomy or adding weight [30, 43].

The modeling of a reconfigurable thruster is not very different from the fixed thruster introduced earlier. A frame $\mathcal{R}_{\boldsymbol{P_i}}$is associated to a thruster and centered on the center of thrust. In this model, the center of thrust is assumed to be static and the vector thruster is considered to rotate around the center of thrust. The direction of the thrust vector is parametrized with two angles: θ_r around $\boldsymbol{y}_{\boldsymbol{P_i}}$ and ψ_r around $\boldsymbol{z}_{\boldsymbol{P_i}}$. Note that the parametrization of the reconfigurable thruster is chosen so that the axis of thrust is aligned with $\boldsymbol{x}_{\boldsymbol{P_i}}$ when $\theta_r = \psi_r = 0$ to keep equivalence with the fixed thruster model introduced earlier. Also, for a 2D reconfigurable thruster, one of the two angles (preferably θ_r) is fixed to 0.

As for the fixed thrusters introduced in section 1.2.1, the effort generated by one reconfigurable thruster at the center of the vehicle are calculated to establish the TCM. In this case though, the force generated by the reconfigurable thruster is not aligned with the $\boldsymbol{x}_{\boldsymbol{P_i}}$ axis as it can be rotated. The force vector is then given in $\mathcal{R}_{\boldsymbol{P_i}}$as:

$$\boldsymbol{F}_{\boldsymbol{P_i}}^{\mathcal{R}_{\boldsymbol{P_i}}} = \begin{bmatrix} \cos\psi_r \cos\theta_r \\ \sin\psi_r \cos\theta_r \\ -\sin\theta_r \end{bmatrix} U_i \tag{1.31}$$

where U_i is the norm of the thrust force of thruster i.

To stick with the model introduced in section 1.2.1 for fixed thrusters, the reconfigurable thrusters will be modeled as three virtual fixed thrusters(respectively, two fixed thrusters for 2DOF reconfigurable thruster) aligned with the axes of $\mathcal{R}_{\boldsymbol{P_i}}$. The force components of the three equivalent fixed thrusters are separated as:

$$u_{x,i} = \cos\psi_r \cos\theta_r U_i \tag{1.32a}$$
$$u_{y,i} = \sin\psi_r \cos\theta_r U_i \tag{1.32b}$$
$$u_{z,i} = -\sin\theta_r U_i \tag{1.32c}$$

Equation (1.32) shows that the three force components generated by a single reconfigurable thruster are independent functions of the thrust norm U_i and the two reconfiguration angles θ_r and ψ_r. This system is invertible. Note that for a 2DOF thruster ($\psi_r = 0$), no force is generated on the $\boldsymbol{y}_{\boldsymbol{P_i}}$ axis. To maintain the equivalence with fixed thrusers, the three force components $u_{x,i}$, $u_{y,i}$ and $u_{z,i}$ of a reconfigurable thruster (respectively $u_{x,i}$ and $u_{z,i}$) are used in the vector of propulsive forces $\boldsymbol{u}$. For a single 3D reconfigurable thruster one gets:

$$\boldsymbol{u} = \begin{bmatrix} u_{x,i} \\ u_{y,i} \\ u_{z,i} \end{bmatrix} \tag{1.33}$$

To show the effects of a 3DOF reconfigurable thruster, a simple *uqr*-craft is introduced. This vehicle is equipped with a single 3DOF reconfigurable thruster at the rear. Figure 1.5 displays this simple example vehicle. The TCM for this vehicle is built as if it was equipped with three fixed thrusters, all placed in the same point $\boldsymbol{P_i}$ and aligned with the axes of the local frame $\mathcal{R}_{\boldsymbol{P_i}}$which, in this case, match the axes of $\mathcal{R}_B$[30]. Table 1.4 shows the propulsive arrangement of this vehicle and the equivalent with fixed thrusters.

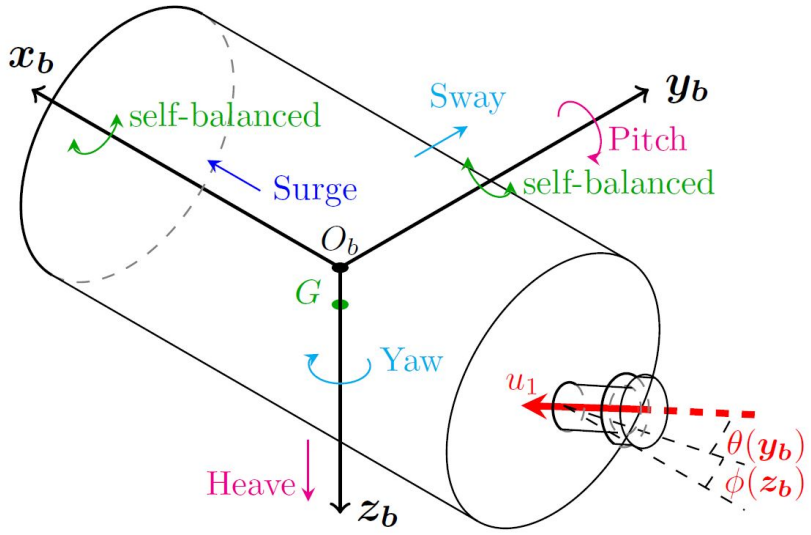


Figure 1.5 – Schematic of the example uqr-vehicle.

Reconfigurable Thruster	**Equivalent virtual fixed thrusters**		
3DOF	**x axis**	**y axis**	**z axis**
$\begin{bmatrix} -L/2 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -L/2 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -L/2 \\ 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -L/2 \\ 0 \\ 0 \end{bmatrix}$
$\theta_{\boldsymbol{P_i}} = 0$	$\theta_x = 0$	$\theta_y = 0$	$\theta_z = -\pi/2$
$\psi_{\boldsymbol{P_i}} = 0$	$\psi_x = 0$	$\psi_y = \pi/2$	$\psi_z = 0$

Table 1.4 – Propulsive configuration of the *uqr* example vehicle

The TCM matrix for this vehicle is then:

$$\boldsymbol{B}_{uqr} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & L/2 \\ 0 & -L/2 & 0 \end{bmatrix} \tag{1.34}$$

Equation (1.34) shows that the uqr-vehicle introduced in this example has three actuated DOF, rank$(\boldsymbol{B}_{uqr}) = 3$. In facts, the reconfigurable thruster positioned at the rear of the vehicle generates a surge force (decoupled from all the other DOF) and two pairs of coupled degrees of freedom: sway and yaw on one side and heave and pitch on the other side. This example allows introducing the compulsory choice between coupled actuated DOF. In facts, when a single actuator —whether it is a fixed thruster or one of the virtual equivalent of a reconfigurable thruster— generates several coupled DOF, one of these DOF must be chosen to be controlled and the other(s) will generate disturbances (See section 5.1 and Appendix C for a more in depth study of the coupled effects of reconfigurable thrusters). In this case, the equivalent fixed thruster aligned with the $\boldsymbol{y}_{P_i}$ axis for instance, generates a sway force coupled with a yaw moment (second column of $\boldsymbol{B}_{uqr}$). In the controller, it will be necessary to choose which DOF is controlled and which is seen as a disturbance. This choice depends mainly on the task and the vehicle. Deeper studies about the consequences of choosing one DOF over the other can be found in section 5.1.

Because of the coupling terms in $\boldsymbol{B}_{uqr}$, the reduction method is used for thrust allocation instead of the pseudo-inverse if the moments are controlled and not the coupled forces. Therefore, the coupled forces generated by the vector thruster are not taken into account in the allocation calculations.

Once $u_{x,i}$, $u_{y,i}$ and $u_{z,i}$ are calculated, the three control parameters of the reconfig-

urable thruster U_i, θ_r and ψ_r are found using the inverse of system (1.32):

$$U_i = \sqrt{u_{x,i}^2 + u_{y,i}^2 + u_{z,i}^2} \tag{1.35a}$$

$$\theta_r = -\text{asin}\left(\frac{u_{z,i}}{U_i}\right) \tag{1.35b}$$

$$\psi_r = \arctan_2\left(u_{y,i}, u_{x,i}\right) \tag{1.35c}$$

Remark 1.2 *The following examples and notably section 5.1 hint towards preferential choices of methods to invert the TCM in the underactuated case and particularly with coupled actuators like off-centered vector thrusters. In facts, in cases where the TCM shows actuator coupling, the Morre-Penrose pseudo-inverse of the TCM cn act as a filter or a gain matrix in front of the control vector. Depending on the coefficients of $\boldsymbol{B}$ and therefore on the physical dimensions of the vehicle, some terms of $\boldsymbol{B}^\dagger$ can tend to zero or on the contrary get very large. Consequently, when working with coupled actuators, the second method based on the reduced TCM is more suited for thrust allocation even though it neglects some phenomena.*

1.2.3 Remark on the effects of coupled actuators

One of the parameters to consider when choosing the thruster technology (fixed or reconfigurable thrusters, control surfaces, ...) and propulsive arrangement of marine vehicles is force and moment coupling. Typically, off-centered thrusters whom force axis does not cross the center of the mobile frame, create a coupled force and moment. This is notably the case for at least one of the fixed equivalent thrusters of a vector thruster. The goal of this section is to demonstrate the effect of this additional disturbing term and compare with an equivalent fixed-thrusters arrangement.

To demonstrate the consequences of this coupling effect, let us compare two surface vehicles. The first one is equipped with two parallel rear thrusters aligned with the $\boldsymbol{x_B}$ axis and the second one is equipped with one 2DOF vector thruster, placed at the rear and rotating around the vertical axis. This two vehicles are reductions of the *RSM* AUV figure 1.4 and the uqr-vessel figure 1.5, respectively, in the horizontal plane. The first one is actuated in surge and yaw and the associated TCM is:

$$\boldsymbol{B}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ -R & R \end{bmatrix} \tag{1.36}$$

Note that in equation (1.36) R can be the radius of the hull of the vehicle if it is approximated to a cylinder or can be half of the distance between the two thrusters if the vehicle is of any other shape. Equation (1.36) shows that surge and yaw are decoupled and no additional force or moment is created by this propulsive arrangement, the two actuators are sharing two DOF without redundancy. This propulsive arrangement can generate a pure yaw moment without coupled effects.

Using the equivalent fixed thrusters introduced in the previous section, the TCM for the second vehicle is:

$$\boldsymbol{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -L/2 \end{bmatrix} \tag{1.37}$$

where L denotes the length of the hull of the vessel. Equation (1.37) shows that the vector thruster of the second vehicle can generate an independent surge force and coupled sway force and yaw moment. The first virtual thruster generates a surge force while the second generates both a sway force and a yaw moment. Following the reasoning introduced in the previous section, it is necessary to choose only one of the two coupled DOF in the control of the vehicle. In this example, yaw will be controlled to keep equivalence between the two vehicles. An analogous example demonstrates the consequences of the other choice in section 5.1.4.

To compare the different behaviors created by these two propulsive arrangements, let us consider a control effort vector $\boldsymbol{\tau}_c$ that would be the output of the controller. The two vehicles are controlled in surge and yaw therefore $\boldsymbol{\tau}_c$ is of shape:

$$\boldsymbol{\tau}_c = \begin{bmatrix} X_c \\ 0 \\ N_c \end{bmatrix} \tag{1.38}$$

where X_c denotes the control surge force and N_c the control yaw moment.

To study the consequence of the coupling terms introduced in the second propulsive arrangement, the real efforts applied to the vehicle are calculated. To do so, the thrust allocation is performed and the resulting efforts expressed in $\mathcal{R}_B$at the center of vehicle are calculated using equation (1.22). For ease of calculation, the two TCM are inverted using the space reduction method introduced in the previous section. Note that the results obtained with the Moore-Penrose pseudo-inverse on $\boldsymbol{B}_2$ vary depending on the numerical values of the model parameters (here L). For coupled actuators, the inversion on the reduced matrix seems more suitable. The control vector is also reduced discarding the sway term: $\boldsymbol{\tau}_c^r = [X_c\ N_c]^\top$. The inverted reduced matrices are:

$$(\boldsymbol{B}_1^r)^{-1} = \frac{1}{2}\begin{bmatrix} 1 & \frac{-1}{R} \\ 1 & \frac{1}{R} \end{bmatrix} \tag{1.39a}$$

$$(\boldsymbol{B}_2^r)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{-2}{L} \end{bmatrix} \tag{1.39b}$$

The real efforts applied to the vehicle in the two cases $\boldsymbol{\tau}_1^*$ and $\boldsymbol{\tau}_2^*$ respectively are then:

$$\boldsymbol{\tau}_1^* = \boldsymbol{B}_1(\boldsymbol{B}_1^r)^{-1}\boldsymbol{\tau}_c^r = \begin{bmatrix} X_c \\ 0 \\ N_c \end{bmatrix} \tag{1.40a}$$

$$\boldsymbol{\tau}_2^* = \boldsymbol{B}_2(\boldsymbol{B}_2^r)^{-1}\boldsymbol{\tau}_c^r = \begin{bmatrix} X_c \\ \frac{-2N_c}{L} \\ N_c \end{bmatrix} \tag{1.40b}$$

Equation (1.40) shows clearly that the thrust allocation and reconstruction of the effort vector creates a new sway force on the second vehicle (second line of $\boldsymbol{\tau}_2^*$). This sway force does not appear on the first vehicle. This phenomena is due to the fact that, with the second vehicle, generating a yaw moment is done by generating an off-centered sway force which also acts as a disturbance. The first propulsive arrangement allows

generation of a pure yaw moment without sway force. Note that the same goes if sway is chosen as a controlled DOF on the second vehicle, a yaw disturbance would appear.

At this point, one might think that this result is counter-intuitive considering that most mainstream surface vehicles actually use a rear vector thruster. In facts, as demonstrated in section 5.1 the consequences of this disturbing sway force can have varying degrees of impact on the vehicle behavior. Notably, the shape of the hull of conventional boats as well as other design parameters make them very robust to sway disturbances, sway motions are highly damped and anti-drift planes stabilize the yaw angle of the vehicle and reduce surge motions at high surge speeds. A more in-depth study on the consequences of these disturbing coupling terms as well as on the model parameters counter-acting these effects can be found in section 5.1.

1.3 Additional remarks on the Modeling and Simulation of Marine craft

This section gives additional remarks on the modeling and simulation of marine craft. Notably, it explains some of the choices made in simulation for the modeling of sea currents, physical limitations of the actuators and measurement noise.

1.3.1 Limitation of reconfiguration speed

To give a more realistic rendition of the behavior of reconfigurable thrusters (also called vector thrusters), the reconfiguration speed has been limited in the simulator. Yet, in order to develop control methods that are independent from technology choices, this limitation will not be taken into account in the following control calculations. Consequently, an unmodeled delay is introduced between the requested control angle and the actual value creating potentially disturbing effects.

Figure 1.6 shows the disturbing force created by the reconfiguration delay in an extreme case ($45°$ error). On figure 1.6 the required force vecotr is represented with the blue arrow and the actual thrust vector is represented with the black arrow. The red arrow represents the disturbing force created by the delay, orthogonal to the desired force.

In order to reduce the norm of the disturbing force in case of reconfiguration delay, the norm of the actual thrust is multiplied by a 0 to 1 parameter calculated as a function

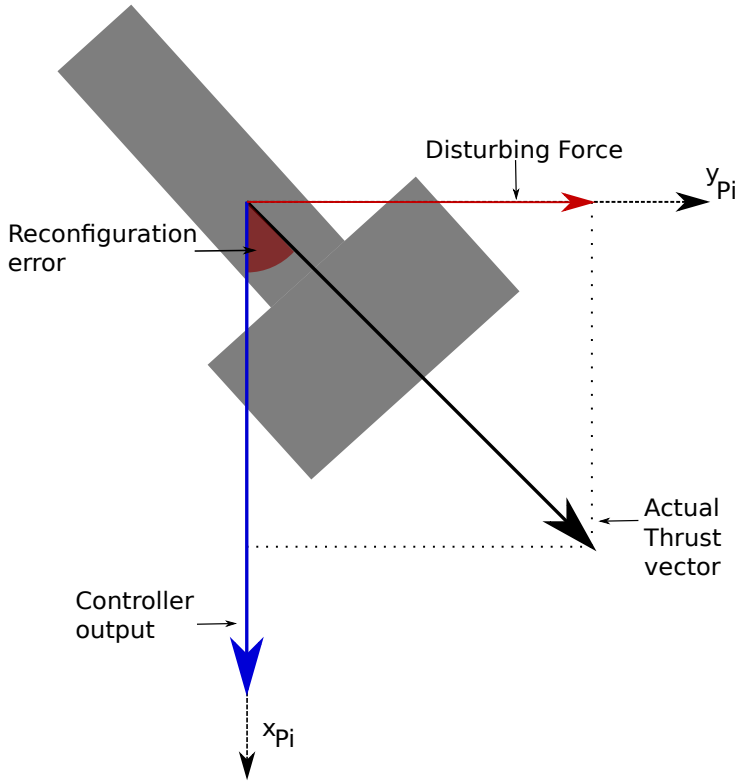


Figure 1.6 – Illustration of the consequences of the reconfiguration delay.

of the reconfiguration angular error. The multiplier is 0 when the reconfiguration error is $\pi/2$ rad or greater, tends to 1 when the reconfiguration error tends to $\pi/4$ rad and is equal to 1 if the error is less than $\pi/4$ rad.

This method reduces the amount of disturbing force created by the reconfiguration delay but also reduces the force component collinear to the controller output. The multiplier barely deviates from 1 practically. Sufficiently smooth trajectories would not require reconfiguration speed higher than the maximum.

1.3.2 Limitation of thrust variation

The simulator used in this work does not model the water surrounding the vehicle. Therefore, it does not simulate the hydrodynamic effects acting on the propellers or the acceleration of water created by the thrusters. Instead, and to avoid unrealistic results, the variation of thrust of the actuators is limited.

This limitation implies a linear thrust profile which also creates a delay in the thruster's response to a control request. This delay is not modeled in the control calculations and therefore mitigates the tracking performance.

1.3.3 Modeling sea currents

Sea currents are the most common external disturbance a marine vehicle is likely to experience. Sea currents are modeled with a constant, irrotational velocity vector (See [64] for more details) defined in $\mathcal{R}_0$ as:

$$\boldsymbol{v_{curr}} = \begin{bmatrix} \dot{x}_{curr} & \dot{y}_{curr} & \dot{w}_{curr} & 0 & 0 & 0 \end{bmatrix}^\top \tag{1.41}$$

and moved to the body-fixed frame $\mathcal{R}_B$ using:

$$\boldsymbol{\nu}_{curr} = \boldsymbol{J}(\boldsymbol{\eta})^{-1}\boldsymbol{v_{curr}} \tag{1.42}$$

The model of the vehicle is then redefined in terms of the relative velocity vector $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_{curr}$. The kinematic model becomes:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})(\boldsymbol{\nu}_r - \boldsymbol{\nu}_{curr}) = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{1.43}$$

and, deriving equation (1.43), the dynamic model becomes:

$$\dot{\boldsymbol{\nu}} = \begin{bmatrix} -\boldsymbol{S}(\boldsymbol{\nu}_2)\boldsymbol{\nu}_{curr} \\ \\ \mathbf{o}_{3x1} \end{bmatrix} + \boldsymbol{M}^{-1}\left(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{g}(\boldsymbol{\eta})\right) \tag{1.44}$$

where $\boldsymbol{S}$ is the skew-symmetrical matrix defined in equation (1.10).

The main advantage of this representation is that the kinematic model of the vehicle is defined only in terms of the absolute velocity vector $\boldsymbol{\nu}$ and the dynamic model is defined in term of the relative velocity vector $\boldsymbol{\nu}_r$. Effects of waves or wind could also be taken into account and modeled as an external force vector translated into a velocity vector linearly added to the current velocity. Then, the amplitude of the current velocity could vary in time in a sinusoidal shape for instance.

1.3.4 Modeling measurement noise

Measurement noise are modeled with a white Gaussian noise. The standard deviation of the noise is arbitrarily set and chosen to represent the reality of common sensors. In the following, the standard deviation for position measures is typically chosen between $0.1\,\mathrm{m}$ and $0.5\,\mathrm{m}$ depending on the experiment and between 0.05 and 0.1 rad for angular measures.

1.3.5 Trajectory generation

This section describes the procedure used to generate trajectories in the simulator. All the DOf of the trajectories must be defined as smooth times functions to be used as references in the control calculations. To fit with all the controller developed in the following (and notably some flatness-based controllers described in chapter 4, the trajectory are continuous and have continuous derivatives up to the fourth order.

In the simulator, each task correspond to one time function. This function describes six sets of lines and curves put together and their derivatives. Each trajectory is discontinuous at that point because continuity is not assured between the different parts of the path.

Then, points of interest are extracted from the trajectories corresponding to each DOF. First points of interest are put down regularly on the trajectory. Then, some more

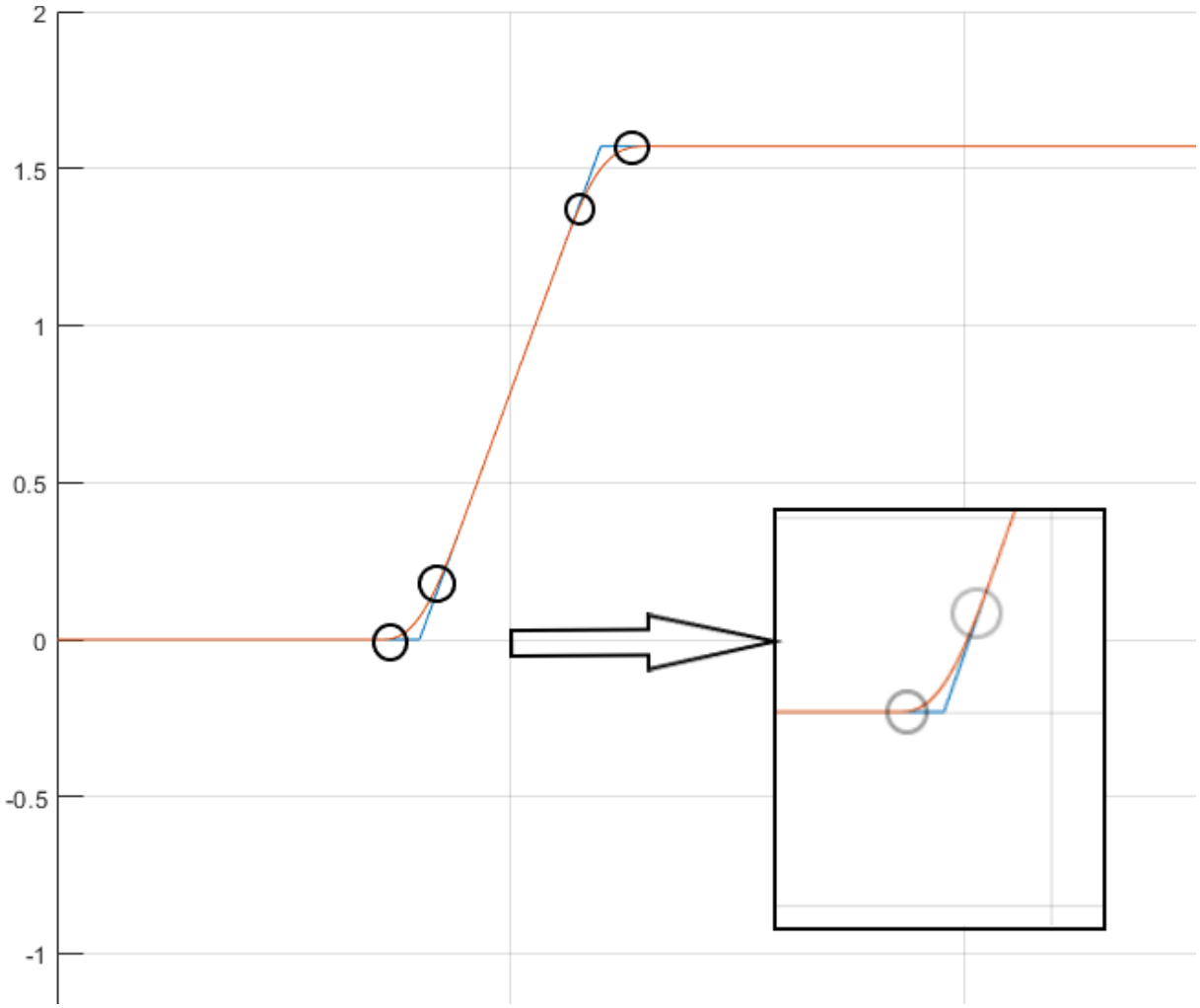


Figure 1.7 – Example of trajectory generation. Blue: discontinuous time function, Black circles: points of interest, Orange: final trajectory

points are added at and around discontinuities, rapid variations in curve, speed or acceleration of the paths. Note that instead of generating discontinuous time functions then extracting interest points, the trajectory could be defined using only sets of waypoints. Previous versions of the simulator were using these time functions as trajectory so they ave been kept to save time.

The final step is to generate new paths between the points of interest. These new segments and curves are calculated using polynomials to ensure continuity and smoothness at and between the points up to the fourth order.

Figure 1.7 shows the result of the trajectory generation procedure on one DOF.

For better trajectory tracking performances, the rtajectory generation procedure

could be upgraded to take information of model and capability of the vehicle into account [130].

1.4 Task description

The first step of task-based design of marine vehicles and controllers is to define (or redefine) the concept of *task*. The task, or application, settles the requirements and constraints to which the vehicle has to conform. Notably, with the aim of reducing the amount of actuators embedded on the vehicle, the evaluation of the number of solicited degrees of freedom (DOF) for a task is of the utmost importance. The notion of task is often confused with the notion of trajectory or path (in the sens of a succession of points in space) the vehicle has to follow. In this work, the definition of a task is extended to take other external phenomena into account. A task is described as the combination of a path, additional constraints, environmental effects acting on the vehicle and natural dynamics of the craft. These four components define the number of required DOF of a task.

Redefining the tasks is the first step towards more maneuverable marine vehicles, allowing for new behaviors and unlocking new applications.

1.4.1 Degrees of Freedom of a task

The trajectory, constraints, environmental effects and natural dynamics of the vehicle define the required DOF for a task.

The trajectory is a set of up to six independent time relative functions, expressed in the inertial frame, describing the expected behavior of the vehicle. These functions represent expected translations in the inertial frame and the expected attitude of the vehicle w.r.t the inertial frame. As a first design base, the vehicle will need to be equipped with actuators allowing generating at least as many translations and as many rotations as defined by the trajectory. Note that translations can be substituted with rotations but not the opposite. In facts, a great number of vehicles (boats, cars, planes, ...) compensate the lack of actuated translations with one or several actuated rotations. The need for a compensation mechanism can make the control problem nontrivial. Controllers exploiting this compensation mechanism are described in chapter 2.

Additionally, the task being most often defined in the inertial earth-fixed frame the

attitude of an underactuated vehicle can modify its degree of actuation with respect to the path and constraints. If only two of the three rotations of a vehicle are actuated for instance, there are sets of possibly stable attitudes in which the vehicle looses the capability of generating a moment around one axis of the inertial frame. As an example, if a vehicle is actuated in roll and yaw but not in pitch, a ninety-degrees roll angle nullifies its ability to turn around the vertical axis of the inertial frame which can be detrimental for the task.

The same goes for translations if all three are not independently actuated. The vehicle can reach attitudes in which it looses its capabilities of generating forces on 1 or several axes of the inertial frame. This phenomenon must be taken into account when the actuator arrangement of the vehicle is designed.

The vehicle may also have to comply with a number of constraints during the application independently of the trajectory. These constraints are not always translatable into path functions. They can be imposed by the design of the vehicle, its software architecture (for instance the pitch singularity using Euler angles), the position of a sensor etc... As an example, for a scanning mission, the vehicle can be equipped with a sonar whom axis must be kept within a 5-degree range of the vertical. Such a constraint can require one or several additional actuators to be filled at all time. For less critical constraints, the designer can also choose to rely on the natural dynamics of the vehicle to meet the constraints like the roll and pitch stability created by an offset between the centers of gravity and buoyancy.

Environmental effects can also be taken into account when defining the task. As an example, a known sea current pushing the vehicle away from the prescribed trajectory can imply adding one or several actuators on the vehicle to counteract it.

Lastly, the natural dynamics of the vehicle can also be considered when defining the task. The shape of the hull, the buoyancy of the vehicle or the relative position of the centers of gravity and buoyancy are all parameters to be taken into account when designing the vehicle for a given task. Depending on the shape of the hull, the vehicle may have one or several energetically inefficient DOF to avoid. In the same way, if the center of mass and the center of buoyancy of the vehicle are offset a constant restauring moment will limit the attitude of the vehicle.

1.4.1.1 Introductory example

Considering all four components of a task (trajectory, constraints, environmental effects, natural dynamics), it appears that designing a vehicle and its propulsive arrangement for a given mission is not always straightforward. It also implies that several solutions will exist for one application. As a first example, for surface vessels, the simple task of following a straight line has two different propulsive arrangement solutions. As a first example the *trajectory* is a 2D segment of the horizontal plane. Considering that the space of this application is composed of two translations and a rotation, two solutions raise for tracking a straight line: two actuated translations or one actuated translation and one actuated rotation.

In the first case, the vehicle would be equipped with two orthogonal fixed thrusters (for simplicity of this example), each of them being able to generate a force moving the vehicle on one axis. In the second case, the vehicle would be equipped with two parallel thrusters which, when combined, can generate a force and a moment independently. From the strict point of view of the number of actuators, these two solutions are equivalent (considering in this example that both vehicles are equipped with same motors and propellers). Assume for now that there is no additional *constraint* or *environmental effects*. Then, the choice between these two solutions is guided by the shape of the hull of the vehicle and its *natural dynamics*. If the vessel has a traditional *V-shaped* hull, it will be much more energetically efficient to choose the second solution over the first one, actuating the favored translation of the vehicle and turning around the vertical axis to point the vehicle in the right direction.

Consider now that the same vehicle is required to maneuver in narrow paths, barely wider than the width of the hull as it could be the case in ports or canal locks. In this new scenario, the additional external *constraint* tips the scale towards the first solution. It would be more practical locally for this application to have two actuated translations and exploit the natural dynamics of the hull for yaw stability even if it is less efficient in regard to energy consumption.

This question is much more complex underwater. To limit the number of embedded actuators, it is often interesting to look for a trade-off between energy efficiency and number of actuators. The designer has to decide whether it is more interesting for the application to locally go against the natural dynamics of the vehicle at the cost of a greater energy expenditure during a short period of time or to add a new thruster and the needed battery and control capacities.

1.4.2 Scenarios

The choice of scenario dictates the time constraint of the task. The three main scenarios for marine craft are *setpoint regulation*, *path following* and *trajectory tracking*. Each scenario corresponds to a level of time constraint.

An application can of course be composed of several scenarios. Taking the example of the surveillance of an offshore submerged power plant, reaching the plant ashore could be a setpoint regulation subtask or a path following subtask but taking images around the plant would likely be a trajectory tracking task.

Tasks can usually be broken down into two distinctive problems: the *geometric* task and the *dynamic* task. The geometric task consists in making the position of one point of the vehicle reach and track a desired target or geometrical path. The dynamic task consists in making the velocity of the vehicle converge to and track the desired velocity assignment. The chosen scenario dictates the importance given to the geometric and dynamic tasks which are achieved either simultaneously or separately.

1.4.2.1 Setpoint regulation

In *setpoint regulation*, the target position and attitude are constant. The behavior of the vehicle on its way to the target is left free. There is no time constraint in this scenario. Several waypoints can be defined and a strategy to switch from one target to the next is chosen. Most often, when the vehicle gets in a prescribed radius of a waypoint, the target changes to the next waypoint. The switching strategy can be adapted to fit the capabilities of the vehicle and smooth its behavior.

Setpoint regulation is notably used with large ships on long missions. A cargo ship crossing the Atlantic Ocean aims for the next port or sets a course to follow for several days at cruising speed. It does not necessarily need to follow a defined path.

The setpoint regulation scenario only focuses on the geometric task of the application. The dynamic task is solved separately and is considered of secondary importance.

1.4.2.2 Path Following

The second scenario is *path following*. In this case, the path taken by the vehicle between the waypoints is constrained. The path can be a straight line between two points or a more complicated shape to avoid obstacles or areas or favor one motion. In path following, the constraints are independent of time, the speed of the vehicle

on the path is not prescribed and the dynamic task is considered solved otherwise. To take the time constraint into account, path following can be updated to path *tracking*. In both cases, the vehicle is aimed for a virtual particle moving along. The dynamics of the particle can be decided a priori to emphasize the time constraint or adapted to the behavior of the vehicle so that the system is never left behind. The work [20, 21] propose a path following and tracking formalism based on the LOS guidance method and show the principle of this type of tasks.

Path following and path tracking applications are notably useful to constrain the vehicle to a path without constraining its speed. To take the example of the cargo ship, the helmsman could choose to follow a known route to reach the next port in which case a path following controller would be recommended. Once the vehicle is on the route it follows the prescribed path and the pilot can choose the desired speed.

The path following scenario emphasizes solving the geometric task. As for setpoint regulation, the dynamic task is considered of lesser importance in path following. The dynamic task can even be sacrificed for the sake of fulfilling the geometric task. This dichotomy gives inherent robustness and flexibility to the path following scheme.

1.4.2.3 Trajectory Tracking

The third scenario and hardest level of time constraint in *trajectory tracking*. Here, the spatial constraints are associated with time requirements. The vehicle is tracking a virtual target moving on a path with a prescribed speed and acceleration. Trajectory tracking usually requires more advanced controllers to maintain good performance. Most of the applications described in this work are trajectory tracking scenarios because it represents the hardest level of difficulty for a vehicle. If the ship is able to track a trajectory at the prescribed speed at all time it will easily be able to track a path with a fixed speed or rally a setpoint. Both setpoint regulation and path following can be replaced by trajectory tracking scenarios with an additional, offline, layer of trajectory planning.

Also trajectory tracking allows smoother and more accurate control of the vehicle all along the task than path following at the cost of higher computational complexity. Trajectory tracking is more suited for applications requiring high degrees of maneuverability. It is notably used with smaller vehicles on applications with shorter time constants.

Trajectory tracking solves the geometric and dynamic tasks simultaneously. The

space and time assignments are mixed up into a single task demanding the vehicle to be in a given point at a given time.

1.5 Definition of Underactuation

This work mainly focuses on the study of marine vehicles described as being underactuated. Underactuation is a well defined concept in robotics in general. In this work a slightly enhanced definition of underactuated systems is proposed to enclose the relation between the system and the task.

In robotics, an underactuated system is usually defined as a system with fewer actuators or control inputs than degrees of freedom. Notably, the work of [160] or [163] give the following definition of underactuation based on Newton's second principle:

Definition 1.3 (Underactuated system) *Considering the system described as:*

$$\ddot{\boldsymbol{q}} = \boldsymbol{f}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{f}_2(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{u} \tag{1.45}$$

where $\boldsymbol{q}$ is the vector of position and attitude, $\boldsymbol{u}$ is a control vector and $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ are two model functions established with Newton's second principle.

The system is considered underactuated if:

$$rank(\boldsymbol{f}_2(\boldsymbol{q}, \dot{\boldsymbol{q}})) < dim(\boldsymbol{q})$$

Definition 1.3 certainly applies to underactuated marine craft considering that $\boldsymbol{q} = \boldsymbol{\eta}$ is the vector of position and orientation of the vehicle. Note that equation (1.45) is equivalent to the representation of the vehicle in the inertial frame even though in marine robotics and with mobile robots in general, the vector of control inputs $\boldsymbol{u}$ is defined in a mobile frame fixed to the body of the vehicle. The actuation matrix $\boldsymbol{f}_2$ must therefore take the rotation of the mobile frame into account.

From a control perspective, it is useful to extend Definition 1.3 to take the required DOF of the task into account. It is notably interesting to introduce the notion of *ill-actuated* vehicle. Following the work of [64], the rank condition on the actuation matrix $\boldsymbol{f}_2$ is redefined in the subspace representing the required DOF of the task. Considering the reduced model:

$$\ddot{\boldsymbol{q}}_r = \boldsymbol{f}_{1,r}(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r) + \boldsymbol{f}_{2,r}(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r)\boldsymbol{u} \tag{1.46}$$

where all lines corresponding to DOF not required in the task are task are discarded in $\boldsymbol{q}_r$, $\boldsymbol{f}_{\boldsymbol{2},r}$ and $\boldsymbol{f}_{\boldsymbol{2},r}$. Then, there are three possible cases:

1. **Underactuated**:
$$\dim(\boldsymbol{q}_r) > \dim(\boldsymbol{u})$$
2. **Ill-actuated**:
$$\dim(\boldsymbol{q}_r) = \dim(\boldsymbol{u}) \;\&\; \text{rank}(\boldsymbol{f}_{\boldsymbol{2},r}(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r)) < \dim(\boldsymbol{q}_r)$$
3. **Fully-actuated**:
$$\dim(\boldsymbol{q}_r) = \dim(\boldsymbol{u}) \;\&\; \text{rank}(\boldsymbol{f}_{\boldsymbol{2},r}(\boldsymbol{q}_r, \dot{\boldsymbol{q}}_r)) = \dim(\boldsymbol{q}_r)$$

In the first case, the vehicle is clearly underactuated with respect to the task. It has fewer actuated DOF than the task requires. This mean that the vehicle will only be able to track part of the task requirements. As a very simple example, if the task is constituted of single setpoint in the horizontal plane, two translations are required. If the vehicle has only one actuated translation, then it is underactuated w.r.t. the task. It is perhaps able to move on one axis to align with the point and meet one requirement of the task but cannot join the point.

The second case is ill actuation. In this case, the vehicle has the same number of actuated DOF as required in the task but they don't match exactly. This is the most common case in marine robotics as it is the case of most surface vessels and many AUVs in position tracking applications. In the horizontal plane, a boat typically has one actuated translation and one actuation rotation (ur-vessel defined earlier). But, if the task requires two translations, typically to reach a point or follow a path defined in positions, the boat misses one translation.

Ill-actuation is particularly interesting from a control perspective because, as seen in many examples of chapter 2, guidance principles and particular designs of the control laws allow exploiting a DOF not required in the task to control one of those missing. Doing so is equivalent to modifying the function $\boldsymbol{f}_{\boldsymbol{2},r}$. Nevertheless, the following examples show that the control of a non-actuated DOF through compensation with another one only works to a certain extend. For instance, ill-actuated vehicles are very hard to control in station keeping tasks because the compensation mechanisms used in the control often rely on kinematic or dynamic effects. In the end, ill-actuated vehicles are

underactuated on the reduced space of the task.

The third and last case is full actuation. In this case the system has as many actuated DOF as requires in the task and they match. In this case the vehicle is able to independently act on every DOF of the task at all times and in all configurations.

CHAPTER 2

LITERATURE REVIEW

Contents

This section presents a literature review on model-based control methods and guidance principles for autonomous marine vehicles. Some of the most common methods found in literature are presented. Detailed examples are given for each of these methods. For each method, a general introduction based on the presentation of a basic example is given before examples in the marine context. This section notably introduces the controllers used throughout this work.

The methods displayed in this chapter are both *guidance* and *control* solutions for fully actuated and underactuated marine craft. There is a traditional dichotomy between guidance and control in the marine literature. This separation seems to be inherited from the early work done on guidance of ships. Usually, *guidance* refers to the algorithm or calculations used on ill-actuated vehicles to turn position error signals into velocity

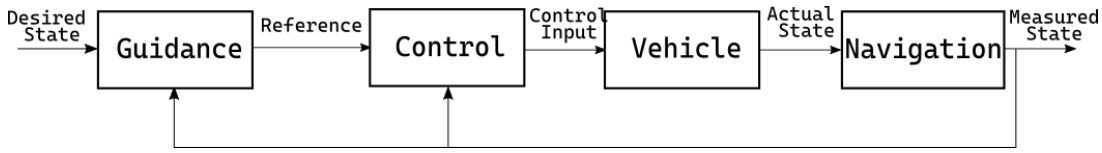


Figure 2.1 – Simplified diagram of the control chain of an AUV

and angle references usable in the *control* of a ship actuated in surge and yaw. A simplified flow chart of the Guidance-Control-Navigation chain is presented on figure 2.1. Yet, as seen in the following examples, guidance and control can be performed simultaneously in some cases. It is notably the case of the Sliding Mode controllers designed for underactuated vessels. They use dynamic coupling relations between the actuated DOF of the vehicle and the DOF that are actually required in the task to perform both guidance and control.

First, this section introduces the most intuitive guidance method for marine vessels: Line of Sight (LOS) guidance [65, 66, 20]. This guidance principle is inherited from naval tradition and mimics the behavior of an experienced boat pilot. Although it is not specific to marine craft, LOS guidance is the go-to method for the most common applications of autonomous marine vehicles. In comparison to the other methods introduced later in this work, LOS is slightly different since it is only a guidance method. Guidance algorithms are used to calculate new sets of references for the vehicle to track instead of the trajectory. The guidance calculations are usually based on the error signals on non actuated DOF and give references for actuated DOF that are not part of the task. Most often with marine vehicles, LOS guidance is used to calculate heading references out of sway errors. The references calculated by the LOS guidance node are then used in commercial controllers.

Then, examples of the different control methods are presented beginning with the PID-based controllers [46, 158]. PID-based controllers are very common when working with linear systems. With nonlinear systems, they are associated with various linearization techniques to perform at best. Some of them are presented and notably model-based feedback, feedforward, and hybrid linearizations. In the underactuated case, examples of additional manipulation allowing compensation of non actuated degrees of freedom are presented. These compensation mechanisms exploit natural relations of the model to compensate the lack of actuation of DOF. Compensation mechanisms act as embedded guidance calculations inside the control law.

The case of feedforward linearization and notably *differential flatness-based* con-

trollers [154, 137] is detailed separately. Differential flatness shows very good performances and robustness when it is applied to nonlinear systems. The few examples of differential flatness application to marine craft show promising results and are presented in more details in chapter 4.

The next control method is Sliding Mode Control (SMC) [175, 34, 87]. SMC is another very common control method for both linear and nonlinear systems offering good robustness to external disturbances and model approximations as well as theoretical finite-time convergence. Note that, unlike PID-based controllers, SMC are constructed to naturaly include model-based linearizing terms. Several formulations exist for SMC as the Terminal Sliding Mode [117] or the Super Twisting Sliding Mode [172], each showing different characteristics. The examples also show that SMC can be applied in the underactuated case and that the construction of the sliding mode control law itself can be tuned to have a guidance role as well.

To complete the study, several adaptive controllers are presented. Notably, the Intelligent PID is described in details as it will be used later in this work. Adaptive methods are not at the heart of this work mostly focused on model-based methods but they are an increasingly notorious part of the literature and their applications show good results.

Additional control methods can be found in [107, 104, 65, 122, 8, 103, 12] among many others. Backstepping, Lyapunov-based methods are also common when working with marine craft as seen in [1] or [102]. We did not retain some approaches that are promising but on which we found very few references, such as [110]. In addition, these last few years, some new control techniques based on machine learning are developed as in [174]. Because they are still recent and not found so often in the literature, these methods are not depicted in this work either.

One of the interests of this section is to see what consequences underactuation may have on the control and how control laws can be designed and adapted to “solve” actuation flaws. To this end, most of the control laws in this section are first presented in the fully actuated case and then underactuated examples are given. Doing so allows for comparing the different strategies used in the underactuated case with the fully actuated case. Some of the control methods presented in this chapter are compared on a prototype vehicle in [164].

2.1 Line of Sight Guidance

The Line Of Sight guidance technique [20, 19, 26, 16] is the most intuitive guidance method for most marine vehicles both on the surface and underwater. The basic idea is rather simple and reproduces the behavior of an experimented helmsman: when piloting a *typical* boat, the easiest and fastest way to reach a distant waypoint is to point the boat towards the waypoint and sail straight forward. Once the waypoint or its neighborhood is reached, the boat is pointed towards the next one and so on.

While being quite simple, the idea behind LOS guidance hides interesting concepts. Inherited from traditional naval techniques, LOS guidance has initially been designed for autonomous boats. Such vehicles are typically ill-actuated in the horizontal plane. Only surge and yaw are actuated on most surface vehicles and sway is passively stabilized by the hull shape. Position tracking tasks in the horizontal plane are thus non-trivial. These vehicles will be referred to as ur*-boats* or ur*-vessels* in the following. There are three main actuation configuration for ur-boats: a fixed rear longitudinal thruster and a rudder, a single reconfigurable rear thruster rotating around the vertical axis or two fixed rear thrusters. These three topologies allow for independently generating a surge force and a yaw moment. The consequences of the choice of actuation arrangement for ur-boats are studied in details in section 5.1. The choice of actuation may have consequences on the control performance of the vehicle. As an example, a fixed thruster and rudder configuration is not suited for on-the-spot rotation or hovering maneuvers. Also, the different possible actuation arrangements of ur-vessels generate different coupled force effects on the sway axis (see section 5.1 for more details).

Setpoint regulation and path tracking are the two most common tasks LOS guidance is used for. The waypoints are usually a set of fixed $(x_d(i), y_d(i))$ points of the horizontal plane. The setpoint regulations and path following scenarios are described in section 1.4.2.

A very simple LOS guidance is given in figure 2.2 or in Section 6.5 of [65]. Here, a new heading reference ψ_d is calculated based upon the position error between the vehicle and the tracked waypoint calculated in the inertial frame. Proportional control allows for tracking of the said heading reference. The surge speed is set to a constant. It could be controlled with another PI-based controller as well. Once the neighborhood of the current waypoint is reached, the next one is targeted. This simple LOS guidance principle does not take the model of the vehicle into account nor does it compensate for

any external disturbances. Intuitively, the heading reference would be calculated as:

$$\psi_d(t) = \arctan_2\left(y_d(i) - y(t), x_d(i) - x(t)\right) \tag{2.1}$$

In (2.1), the $\arctan_2$ function is used instead of the classical atan function to avoid singularities. This expression is commonly used in numerical calculation; it extends the definition of atan to the complete complex plane. The $\arctan_2$ function is defined on the four quadrants of the complex plane as:

$$\begin{array}{rcl} \mathbb{R}^2 \setminus (0,0) & \longrightarrow & [-\pi; +\pi[\\ (y,x) & \longmapsto & \arctan_2(x,y) = \begin{cases} +\pi + \arctan\left(\frac{y}{x}\right) & \text{if } x < 0, y > 0 \text{ (quadrant II)} \\ +\frac{\pi}{2} & \text{if } x = 0, y > 0 \\ \arctan\left(\frac{y}{x}\right) & \text{if } x > 0 \text{ (quadrants I and IV)} \\ -\frac{\pi}{2} & \text{if } x = 0, y < 0 \\ -\pi + \arctan\left(\frac{y}{x}\right) & \text{if } x < 0, y < 0 \text{ (quadrant III)} \\ -\pi & \text{if } y = 0, x < 0 \end{cases} \end{array}$$

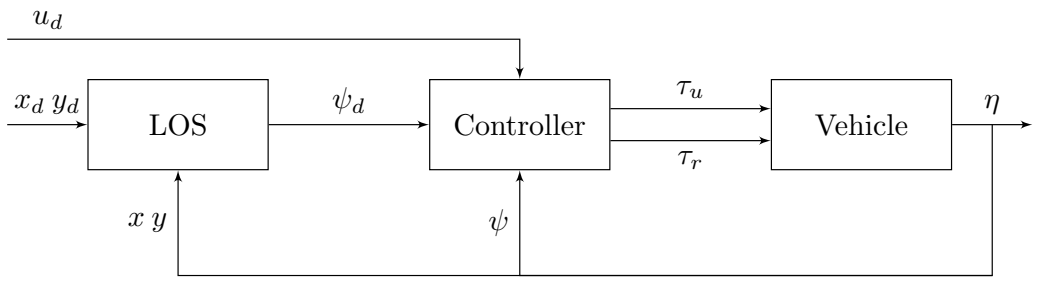


Figure 2.2 – Block diagram of a simple Line Of Sight Guidance control. The *Controller* block represents any control function capable of calculating the surge and yaw controls $\boldsymbol{\tau_u}$ and $\boldsymbol{\tau_r}$ from the desired and current states.

A more advanced control based on LOS guidance is given in [66]. In this work, LOS guidance algorithms provide desired heading and its first and second order derivatives. The nonlinearities of the model are taken into account and linearizing terms are added in the controller in a hybrid feedback and feedforward fashion (see Section 2.2 for more

details on linearizing controllers). The two controller equations can be expressed as:

$$\boldsymbol{\tau_c} = \begin{bmatrix} X_c \\ 0 \\ N_c \end{bmatrix} \tag{2.2a}$$

$$X_c = m_{11}\dot{u}_d + n_{11}u - k_1(u - u_d) \tag{2.2b}$$

$$N_c = m_{32}\dot{v} + m_{33}\dot{r}_c + n_{32}v + n_{33}r - k_3(r - r_c) - (\psi - \psi_d) \tag{2.2c}$$

$$r_c = -c(\psi - \psi_d) + r_d \tag{2.2d}$$

where m_{ij} are coefficients of the mass matrix, n_{ij} are coefficients of the Coriolis and damping matrix, k_1, k_3 and c are control gains. The yaw speed control r_c is calculated as an intermediate command variable in the kinematic controller (2.2d). In the controller given by (2.2a), the heading references ψ_d, r_d and $\dot{r}_d$ are outputs of an extended LOS guidance algorithm similar to (2.1).

LOS guidance can also be applied in the underwater three-dimensional case as displayed in [20]. Underwater, LOS guidance is mainly used for vehicles actuated in surge, pitch and yaw called uqr-*vessels* in the following. For uqr-vessels, pitch and yaw are generally used to cope for the lack of sway and heave actuation therefore allowing tracking of (x, y, z) waypoints or trajectories. The same principle as for 2D LOS guidance is applied, the vehicle is pointed towards the tracked point and is propelled forward. As shown in [20], 3D LOS guidance involves an additional angle: the elevation. As in the 2D case, the heading and elevation angles given by LOS guidance can be used as references in the yaw and pitch controllers respectively.

The work in [20] provides three assumptions for stability of the LOS guidance-based controller and also makes use of two additional control parameters called *look-ahead distances*. When look-ahead distances are used, the ship is not pointed towards the target itself but towards a fictional point usually chosen further away on the trajectory. Look-ahead distances allow a smoother trajectory convergence and can be tuned relatively to the application and system. This work also introduces a path frame $\boldsymbol{R_P}$ centered on the tracked point p. Conditions upon the evolution of point p are given in this work to ensure convergence in the form of an equation giving the evolution speed of point p relatively to the desired speed of the vehicle. The tracking errors used for the calculation of the reference elevation and heading angles of the LOS algorithm are

calculated in frame $\boldsymbol{R_P}$. The work of [20] is actually quite different from the other examples of LOS guidance of this section. It is studied in more details and expanded in section 3.5.2.

In [19], the same authors apply the principle of 3D LOS guidance to a uqr-ship and build a complete controller upon this principle. As seen before, the LOS angles are used as references in the pitch and yaw controllers. This work also gives a lead towards unification of fully actuated and underactuated controllers, taking into account the fact that some actuated DOF may be considered non-actuated in certain speed ranges. In addition, [26] gives experimental results of a similar control method in the case of an underwater vehicle tracking a predefined fixed-depth (x_d, y_d) path close to the surface.

The LOS guidance principles described up to this point do not take external disturbances like marine current, wind or waves into account. In fact, traditional LOS guidance does not ensure theoretical convergence in the presence of external disturbances. To cope for such disturbances, [16] proposes adding a new term in the traditional LOS heading angle calculation in the case of a ur-vessel. This new term, denoted as y_{int}, behaves like the integral term of a PI controller. It cancels possible steady state error due to persistent external disturbances. Though y_{int} is not calculated as the integral of y, its propagation function will be chosen to allow convergence of the closed-loop system. Therefore, y_{int} will be referred to as a *pseudo-integral* term. Considering $y_d = 0$, the modified LOS heading angle is:

$$\psi_d = \operatorname{atan}\left(\frac{y + \sigma_y y_{int}}{\Delta}\right) \tag{2.3a}$$

$$\dot{y}_{int} = \frac{\Delta y}{(y + \sigma_y y_{int})^2 + \Delta^2} \tag{2.3b}$$

In (2.3), σ_y is a new control parameter acting as an integral gain, and Δ is the look-ahead distance. As Δ is always strictly positive, RHS of Equation (2.3) is always defined, and the value of ψ_d is always in $[-\frac{\pi}{2}, +\frac{\pi}{2}]$. For a complete four quadrants output, one may use the $\arctan_2$ function instead.

Equation (2.3b) gives the propagation rate of the *integral* term y_{int}. The first order derivative of the pseudo-integral term y_{int} is conveniently chosen to allow convergence of the closed-loop system.

In [16], the tracked trajectory is a straight-line path defined in $\boldsymbol{R_0}$ by $y_d = 0$. Therefore, the position of the vehicle y used in (2.3) could be replaced by the cross-track error for different trajectories. Nonetheless, having the integral term y_{int} in (2.3) allows the

vehicle to move along the path $y = 0$ with a non-zero relative heading angle in case of external disturbances. An adaptive yaw controller is then given and stability is proven in the presence of external disturbances.

The work presented in [24, 25] provides a generalization of the integral LOS guidance in 3D in the presence of sea currents. In the former, the addition of an integral term in the elevation angle calculation allows for compensation of vertical oceanic current in the case of a horizontal trajectory tracking problem for a uqr-ship while, in the latter, both elevation and heading angles are given with an integral term therefore allowing robustness to any irrotational current. The example in [25] is interested in tracking the x-axis line defined with $y_d = z_d = 0$. The enhanced heading angle in [25] is similar to (2.3), and the elevation angle is built the same way but with the z tracking error:

$$\theta_d = \text{atan}\left(\frac{z + \sigma_z z_{int}}{\Delta_z}\right) \tag{2.4a}$$

$$\dot{z}_{int} = \frac{\Delta_z z}{(z + \sigma_z z_{int})^2 + \Delta_z^2} \tag{2.4b}$$

In [69, 106], a slightly different approach of LOS guidance is proposed. Here, the calculation of the desired LOS heading angle is based upon the cross-track error. The cross-track error kinematics is given by:

$$\dot{y}_e = U \sin\left(\psi - \psi_p(s) + \beta\right) \tag{2.5a}$$

$$\psi_p(s) = \arctan_2(y_p'(s), x_p'(s)) \tag{2.5b}$$

where $U = \sqrt{u^2 + v^2}$ is the velocity of the ship, $\psi_p(s)$ is the trajectory heading angle, and $\beta = \arctan_2(v, u)$ is the crab angle of the vehicle. The variable s can be considered as a curvilinear abscissa whom propagation rule is given in [69]. More details about the path frame used here can be found in the references and in section 3.5.2.

Thus, the cross-track error kinematics (2.5a) can be seen as a new tracking problem of input $\Psi = \psi + \beta$ and output y_e, where Ψ is the course angle. A new formulation of the desired LOS heading angle is given to stabilize the cross-track error towards the equilibrium point $\dot{y}_e = 0$; the so-called *proportional* LOS guidance:

$$\Psi_d = \psi_p(s) + \text{atan}\left(-\frac{y_e}{\Delta}\right) \tag{2.6}$$

It is worth noting that [69] proposes an alternative representation of the problem.

A *pivot* point is introduced and is chosen as the point of the vehicle where the local sway velocity is zero. The pivot point is considered as the new tracking point. A new definition of the cross-track error and of the tracking problem in general are given in this point. The proportional LOS guidance is demonstrated as being uniform semiglobal exponentially stable (USGES) for the ur-boat. Surge and yaw controllers are given in this case based on the cross-track error. In addition to minimizing the cross-track error, the work of [106] also includes minimization of the along-track error with a surge speed controller taking both cross-track and along-track errors into account.

Some recent work conducted in [62] and [61] pushes the concept of integral LOS guidance forward with a new adaptive LOS algorithm showing increased robustness to unknown current. Also, this work uses a new amplitude-phase representation of the kinematic model of the vehicle. These adaptive solutions are described along-side other adaptive controllers in section 2.5.

LOS guidance has received a lot of attention over the years and many more references could be added in this work. The method has been used in different applications such as waypoint tracking control in [132], studies on the optimization of the look-ahead distance choice have been conducted as in [106], and LOS guidance has been applied to smooth transitions between fully actuated and underactuated configurations in [173] and, up to this date, more work is conducted on the application of LOS guidance to different marine craft [68, 111]. Also, [105] gives a good overview of LOS guidance in 2D and 3D proposing both independent horizontal and vertical planar solutions as well as a different solution coupled with an auto-depth controller.

Overall, LOS guidance can be considered as the go-to method for automation of ur-vessels in surface or planar applications and uqr-ships in underwater applications. However, as can be seen here, LOS guidance is not suited for applications where orientation of the vehicle is controlled or in hovering tasks. In fact, LOS is one of the methods using a rotational DOF to compensate the lack of actuation on a translation. In the ur-ship case, yaw moment is used to compensate for the lack of sway. However, the difference between LOS guidance and the other compensation methods given later in this work is that, with LOS guidance, the compensation occurs at the guidance level while, in the following, it happens mostly at the controller level. Using two translation errors or two translation velocities in the calculation of a reference for a rotational DOF makes the controller on the latter a function of either of these translation signals. This kind of manipulation is useful for underactuated systems since it allows going beyond

the traditional one-to-one diagonal controller systems where control over a DOF is calculated upon the error on this same DOF. Other methods of this kind are presented in the following.

2.2 Model-Based Linearization and PID Control

The very early works in (static state) feedback linearization can be found in W. Korobov [99] and R.W. Brockett [22]. The necessary and sufficient conditions of feedback linearization have been obtained by B. Jakubczyk and W. Respondek [94]; see the works of R. Su and A. J. van der Schaft [162, 145] for the general extension to the nonlinear case. Refer to D. Claude [32] for a survey. The problem of dynamic feedback linearization has been later addressed by B. Charlet, J. Lévine and R. Marino [27]. See also the books [91, 126, 156] for more references. Recall that the problem of dynamic state feedback linearization is still open. The input–output linearization problem has been first addressed in [92] and completely solved in an algebraic setting in [152].

Proportional Integral Derivative (PID) control is the most well-known and widely spread control technique among autonomous systems. However, when it comes to marine craft and nonlinear systems in general, PID control itself may not be enough to cancel the state error in trajectory tracking tasks. Additional linearizing mechanisms must be associated with PID control when working with nonlinear systems such as autonomous boats or underwater vehicles. This section demonstrates the use of PID control and these additional strategies in the case of marine craft. The main advantage of adding linearizing terms to the control law is to create a linear or quasi-linear closed-loop system by canceling the nonlinearities of the model, whereas a PID controller alone applied to a nonlinear system would be particularly difficult to tune and concluding on the convergence and stability of the closed-loop system would not be possible.

This section displays examples of PID-based controllers both in the fully actuated case and in the underactuated case. Note that, in the fully actuated case, the common use of diagonal gain matrices creates a 1-to-1 relation between input and output. On the other hand, in the underactuated case, additional non-diagonal mechanisms create compensation behaviors used to control a non-actuated degree of freedom with another actuated one through natural model relations.

This section mainly focuses on model-based linearization methods but model-free, adaptive controllers also exist and some examples of such controllers can be found

in [65, 158, 118, 15] and in section 2.5.

Linearizing model-based controllers can be broken down into three classes. The first type is *State Feedback Linearization*, often referred to as *Exact* Linearization [65, 118]. Here, components of the model evaluated at the current state of the system are used in the controller in order to theoretically exactly cancel the nonlinearities in the closed-loop system. Practically, some nonlinear terms may appear in the closed loop system depending on the experimental conditions and approximations. This methods is especially sensitive to model approximation and variation in experimental conditions introducing unexpected nonlinear terms in the closed-loop system. Nonetheless, the main interest of using exact linearization is turning the original nonlinear control problem into a linear (or almost linear) closed-loop system in which conventional PID tuning methods as pole placement or linear quadratic regulation can be used.

The second class of linearizing model-based controllers can be referred to as *Feedforward Linearization* in opposition to the first one but is mostly called *Flatness-based* control as a reference to the flat characteristics of the system. Flatness-based control is discussed in details in the section 2.3 and is investigated deeply in chapter 4. In this case, the model parameters added to the controller are evaluated at the desired state or at a virtual reference. Therefore, as long as the current state is different to the desired state or the reference used in the feedforward terms, the nonlinear terms of the system are not exactly canceled in the closed-loop system. The resulting state error dynamics can therefore remain nonlinear. In this second case, conventional tuning methods are therefore more difficult to set up on the resulting closed-loop system. The non-linear terms cancel out around the desired state or reference.

The third class of linearizing model-based controllers is hybrid between feedback and feedforward linearization [46, 65, 158, 118]. In these controllers, both feedback and feedforward terms are used to compensate the nonlinearities of the system either everywhere or only in the vicinity of the trajectory. Such controllers can be used to bring some “well-behaving” nonlinear terms in the closed-loop system, therefore enhancing the overall performances. Outside of the marine context and with other types of controllers, details about these nonlinearities of the closed-loop system and their interest can be found in [84]. Some of the examples introduced in this section allow for comparing hybrid linearizing controllers using different amounts of feedforward.

As will be detailed later in this section, it is frequent to use a virtual reference in the controller which may be apprehended as having two nested control loops or two stages.

As a simple example, the virtual reference in the case of marine craft would be a virtual velocity vector built itself as a controller assuring convergence of the position of the vehicle towards the trajectory. Broadly, the outer loop generates an effort control vector ensuring convergence of the vehicle's velocity towards the virtual reference. This virtual reference is calculated as a velocity controller, often PI-based, assuring convergence of the position towards the trajectory. Using such a two-staged architecture is notably useful to include the guidance principle in the controller. Most often, the guidance or compensation mechanism is included in the first stage or outer loop of the controller.

This type of two-staged systems are often called *cascaded* systems [115]. Cascaded systems are used in numerous control applications inside and out of the marine context. The cascade structure provides a natural robustness and facilitates tuning of the controller.

2.2.1 Examples of linearizing controllers

This section demonstrates the three different linearizing controllers introduced earlier based on an example of [156]. In the original reference, a feedback linearizing controller is used.

Let us introduce a nonlinear system of state x and input u. For simplicity, the system used here is of scalar state and input:

$$\dot{x} = f(x) + g(x)u(t) \tag{2.7}$$

The functions $f(x)$ and $g(x)$ are known nonlinear functions of the state x and $g(x)$ is assumed to be non-singular.

Three different PI-based linearizing controllers are given for this example following the four categories introduced earlier. table 2.1 associates one controller with its usual name.

$u_1(t)$	Feedback Linearization
$u_2(t)$	Exact Feedforward Linearization (Flatness)
$u_3(t)$	Hybrid Linearization

Table 2.1 – Names of linearizing controller categories

$$u_1(t) = g(x)^{-1}\left(\dot{x}^* + \text{PI}(x^*, x) - f(x)\right) \tag{2.8a}$$
$$u_2(t) = g(x^*)^{-1}\left(\dot{x}^* + \text{PI}(x^*, x) - f(x^*)\right) \tag{2.8b}$$
$$u_3(t) = g(x)^{-1}\left(\dot{x}^* + \text{PI}(x^*, x) - f(x^*)\right) \tag{2.8c}$$

In equation (2.8), the $*$ denotes a desired value or a reference calculated otherwise. A PI controller is used in this example but note that other control laws could be used instead to close the loop.

For comparison, the three corresponding closed-loop systems are given:

$$\dot{x} = \dot{x}^* + \text{PI}(x) \tag{2.9a}$$
$$\dot{x} = f(x) + g(x)g(x^*)^{-1}\left(\dot{x}^* + \text{PI}(x) - f(x^*)\right) \tag{2.9b}$$
$$\dot{x} = \dot{x}^* + \text{PI}(x) - \left(f(x^*) - f(x)\right) \tag{2.9c}$$

Equation (2.9) shows the three closed-loop systems respectively corresponding to the three controllers of equation (2.8) applied to the system (2.7).

The consequences of the different linearizing solutions appear clearly in equation (2.9). The feedback linearizing controller equation (2.8a) leads to the fully linear closed-loop system equation (2.9a). All non-linear terms are exactly canceled provided that the measured state and model terms used in the calculations of $u_1(t)$ are exact.

The feedfoward linearizing controller equation (2.8b) gives non-linear closed-loop systems. Note however that as the system tends to the desired state, the remaining non-linear terms tend to cancel out and the closed-loop system (2.9c) becomes similar to equation (2.9a). Of course the feedforward controller operates at best when the vehicle is in the vicinity of the desired trajectory. The behavior is jeopardized when the error between the desired and actual state of the vehicle gets too large. Consequently, feedforward linearizing controllers need to be used in association with a re-planning strategy recalculating the desired trajectory if the system gets too far from the path or in applications where the system is ensured to stay in an acceptable neighborhood of the trajectory.

For the hybrid linearizing controller equation (2.8c), measured values are used in $g(x)^{-1}$ while desired values are used in $f(x^*)$. This choice leads to the closed-loop system (2.9c) where an additional error term $(f(x^*) - f(x))$ appears. This term can enhance

the stability of the system as a damping term or be damageable. The choice of using feedforward or feedback for each term of the model must ensure that all the terms not exactly canceled and remaining in the closed-loop are actually favorable to the convergence of the system and not damaging for the tracking mission. In the vicinity of the trajectory, these terms are canceled eventually.

These control structures can be identified in the examples introduced in the following and more details are given in the cited works and in the following sections as for the interest of using one linearizing method over the other.

Obviously, such linearizing controllers are not only used with marine vehicles. They can be applied to many nonlinear systems as in [128, 125]. Linearizing controllers are applied to a generic dynamic system in the former and to a manipulator arm in the latter. These two references refer to the methods used as the *Computed torque*. Although it is very close to the linearization methods used in the following examples, the term "computed torque" is rarely used when working with marine craft. In [17], computed torque control is used to anticipate and control the behavior of an eel-like robot.

2.2.2 Application to Fully Actuated Vehicles

2.2.2.1 Feedback Linearizing Controller

An example of a PID-based feedback linearizing controller can be found in [65] (in Sections 7 and 14). The closed-loop system obtained when applying the controller is linear. The control law is given by:

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{2.10a}$$

$$\dot{\boldsymbol{\nu}}_r = \boldsymbol{J}(\boldsymbol{\eta_2})^{-1}\ddot{\boldsymbol{\eta}}_r - \dot{\boldsymbol{J}}(\boldsymbol{\eta_2})^{-1}\dot{\boldsymbol{\eta}} \tag{2.10b}$$

$$\ddot{\boldsymbol{\eta}}_r = \ddot{\boldsymbol{\eta}}_d + \boldsymbol{K_D}\dot{\boldsymbol{e}}_\eta + \boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_I}\int_0^t \boldsymbol{e}_\eta(\zeta)d\zeta \tag{2.10c}$$

In (2.10), the acceleration reference $\ddot{\boldsymbol{\eta}}_r$ is built as a typical PID controller associated with the acceleration feedforward term $\ddot{\boldsymbol{\eta}}_d$. This control law is very close to the example demonstrated in section 2.2.1. It is clear that applying the control law (2.10) leads to a linear closed-loop system dynamics. In fact, because they are evaluated at the actual state of the system, the nonlinearities of the model $\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta})$ are exactly

canceled by the control law. The closed-loop system is now:

$$\dot{\boldsymbol{\nu}} = \dot{\boldsymbol{\nu}}_r \quad \Rightarrow \quad \ddot{\boldsymbol{\eta}} = \ddot{\boldsymbol{\eta}}_d + \boldsymbol{K_D}\dot{\boldsymbol{e}}_\eta + \boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_I}\int_0^t \boldsymbol{e}_\eta(\zeta)d\zeta \tag{2.11}$$

The exact feedback linearization used in this example allows global exponential convergence of both the velocity and position of the vehicle. This example also highlights another advantage of the exact linearization which is that it allows using conventional linear gain tuning methods on the closed-loop system.

This controller can be seen as a *two-staged* controller. The first stage calculates the acceleration reference $\ddot{\boldsymbol{\eta}}_r$ and the second stage uses this reference to calculate the control vector $\boldsymbol{\tau}$ and cancel the non-linear part of the model. This structure is commonly used in non-linear control as it makes a very understandable and predictable behavior of the system. It is notably useful when working in two different frames rotating one with respect to the other as it is the case here. The acceleration reference is computed in the inertial frame and expressed in the body-fixed frame where it can be used to calculate the force and moment vector as in equation (2.10a).

This control method is used throughout this work as a reference PID-based control method for fully actuated vehicles. Performances of such a controller are detailed in section 3.3.

2.2.2.2 Hybrid Linearizing Controllers

A first example of a hybrid linearizing controller can be found in [46]. In this work, the model is considered in the inertial frame and an expression of the model matrices expressed in this frame can be found in the reference. The modified matrices in the inertial frame are indicated with $\bar{\cdot}$. The controller is given by the set of equations:

$$\boldsymbol{\tau} = \bar{\boldsymbol{M}}\ddot{\boldsymbol{\eta}}_r + \bar{\boldsymbol{C}}\dot{\boldsymbol{\eta}}_r + \bar{\boldsymbol{D}}\dot{\boldsymbol{\eta}}_r + \bar{\boldsymbol{g}}(\boldsymbol{\eta}) + \boldsymbol{\Lambda}\boldsymbol{\epsilon} \tag{2.12a}$$

$$\dot{\boldsymbol{\eta}}_r = \boldsymbol{K_D}\dot{\boldsymbol{\eta}}_d + \boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_I}\int_{t_0}^t \boldsymbol{e}_\eta(\zeta)d\zeta \tag{2.12b}$$

$$\boldsymbol{\epsilon} = \boldsymbol{K_D}\dot{\boldsymbol{e}}_\eta + \boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_I}\int_{t_0}^t \boldsymbol{e}_\eta(\zeta)d\zeta \tag{2.12c}$$

In Equation (2.12), the orientation of the vehicle is represented in quaternions as part of the vector $\boldsymbol{\eta}$, and $\boldsymbol{e}_\eta$ is the state error. The matrices $\boldsymbol{K_D}$, $\boldsymbol{K_P}$, $\boldsymbol{K_I}$ and $\boldsymbol{\Lambda}$ are strictly definite positive gain matrices that are set to the identity in this work but could be tuned

for better performances. The reference [46] specifies that removing the integral term by setting $\boldsymbol{K_I}$ to 0 does not disturb the global convergence of the method. A similar approach can also be found in [77] where the saturation is taken into account in the controller.

In the control law (2.12), $\boldsymbol{\epsilon}$ is built as a conventional PID controller outputting an acceleration vector and $\dot{\boldsymbol{\eta}}_r$ is a virtual velocity reference as explained at the beginning of this section. This reference can be seen as a kinematic controller assuring convergence of the position of the vehicle towards the trajectory. It is interesting to note that, when the state error tends to zero, the virtual velocity reference $\dot{\boldsymbol{\eta}}_r$ tends to the desired speed in the inertial frame multiplied by a control parameter $\boldsymbol{K_D}$.

Removing the integral terms of the controller, the closed-loop error system can be expressed in the inertial frame as:

$$0 = \boldsymbol{M}^*(\boldsymbol{K_D}\ddot{\boldsymbol{\eta}}_d - \ddot{\boldsymbol{\eta}}) + (\boldsymbol{C}^* + \boldsymbol{D}^*)(\boldsymbol{K_D}\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{\eta}}) + \mathcal{K}_\mathcal{D}\dot{\boldsymbol{e}}_\eta + \mathcal{K}_\mathcal{P}\boldsymbol{e}_\eta \tag{2.13a}$$

$$\mathcal{K}_\mathcal{D} = \boldsymbol{M}^*\boldsymbol{K_P} + \boldsymbol{\Lambda}\boldsymbol{K_D} \tag{2.13b}$$

$$\mathcal{K}_\mathcal{P} = (\boldsymbol{C}^* + \boldsymbol{D}^*)\boldsymbol{K_P} + \boldsymbol{\Lambda}\boldsymbol{K_P} \tag{2.13c}$$

Because of the hybrid linearization used in this example, the closed-loop dynamics (2.13) is nonlinear, the Coriolis, centripetal and damping terms are not canceled. Note that, new ambiguous error terms $(\boldsymbol{C}^* + \boldsymbol{D}^*)(\boldsymbol{K_D}\dot{\boldsymbol{\eta}}_d - \dot{\boldsymbol{\eta}})$ are created in the closed-loop system because of the hybrid character of this controller. However, the reference [46] states that the system is globally asymptotically convergent but it can hardly be seen on the closed-loop system equations.

Similar examples of nonlinear PD controllers for trajectory tracking are demonstrated in Sections 7 and 14 of [65]. Here, the control laws are given in the mobile frame as:

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\boldsymbol{K_D}\boldsymbol{\epsilon} \tag{2.14a}$$

$$\boldsymbol{\epsilon} = \dot{\boldsymbol{e}}_\eta + \boldsymbol{\Lambda}\boldsymbol{e}_\eta \tag{2.14b}$$

$$\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}}_d + \boldsymbol{\Lambda}\boldsymbol{e}_\eta \tag{2.14c}$$

$$\boldsymbol{\nu}_r = \boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}}_r \tag{2.14d}$$

Again, $\dot{\boldsymbol{\eta}}_r$ (and thus $\boldsymbol{\nu}_r$) can be considered as the virtual velocity reference used in the feedforward part of the control law, $\boldsymbol{e}_\eta = \boldsymbol{\eta}_r - \boldsymbol{\eta}$ and $\boldsymbol{e}_\nu = \boldsymbol{\nu}_r - \boldsymbol{\nu}$. Note that in this

example as in the previous the feedforward and feedback terms are nonlinearly mixed together as in $\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu}_r$. Some of the following hybrid controller examples show more of a linear combination of linear and nonlinear terms leading to simpler closed-loop systems. Here, nonlinearities will remain in the closed-loop system. However, it should be noted that, once the position error is canceled ($\boldsymbol{e}_\eta = 0$), the velocity reference $\dot{\boldsymbol{\eta}}_r$ is equal to the desired velocity $\dot{\boldsymbol{\eta}}_d$. Therefore, when the vehicle tends towards the desired state, the reference terms tends to the desired values and the controller behaves like a common feedforward linearizing controller.

More examples of hybrid linearizing controllers can be found in [158]. This work proposes and compares a set of controllers using either a hybrid linearization with both feedforward and feedback terms, only feedforward terms or adaptive structures. The model-based controllers introduced in [158] show interesting results. First, a conventional PD controller that does not rely on the model, is given as a baseline for comparison. The PD control law is given as:

$$\boldsymbol{\tau} = \boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_D}\boldsymbol{e}_\nu \tag{2.15}$$

As said earlier in this section, a simple PD controller is unlikely to show good performances when used to control a marine craft on complex trajectory tracking tasks. However, it is worth mentioning here because it can be a first step towards an autonomous vehicle and give results on simple position keeping task.

Note that, in [158], the system is represented with a decoupled model. All non-diagonal terms are neglected and notably the non-diagonal added masses and Coriolis and centripetal terms. Therefore, most of the nonlinearities of the model are neglected. The model can therefore be considered as six independent nonlinear subsystems, one per degree of freedom. In addition, the linear and quadratic damping terms are broken apart and regrouped in two different matrices, respectively, $\boldsymbol{D_L}$ and $\boldsymbol{D_Q}$. The first control law is given as:

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\nu}}_d + \boldsymbol{D_Q}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D_L}\boldsymbol{\nu}_d + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_D}\boldsymbol{e}_\nu \tag{2.16}$$

The control law (2.16) is hybrid in the sense that it mixes feedforward and feedback terms. The control law can be broken down into three parts. First, one finds a traditional PD controller similar to the baseline PD control law (2.15): $\boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_D}\boldsymbol{e}_\nu$. Matrices $\boldsymbol{K_D}$ and $\boldsymbol{K_P}$ are usual gain matrices. Then, the linear part of the model is added and

evaluated at the desired state, that is: $\boldsymbol{M}\dot{\boldsymbol{\nu}}_d + \boldsymbol{D_L}\boldsymbol{\nu}_d$. Finally, the nonlinear terms of quadratic damping, gravity, buoyancy and disturbance effects are added and evaluated at the actual state: $\boldsymbol{D_Q}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta})$. This last part is exactly canceling the nonlinear part of the system.

Now, let us introduce the second control law of [158] for the sake of comparing the closed-loop systems they both lead to. This second control law is mostly similar to the first one, but this time feedforward is used in the quadratic damping. The second control law is then:

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\nu}}_d + \boldsymbol{D_Q}(\boldsymbol{\nu}_d)\boldsymbol{\nu}_d + \boldsymbol{D_L}\boldsymbol{\nu}_d + \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{K_P}\boldsymbol{e}_\eta + \boldsymbol{K_D}\boldsymbol{e}_\nu \tag{2.17}$$

Note that exact feedback values of the nonlinear gravity, buoyancy and disturbance term $\boldsymbol{g}(\boldsymbol{\eta})$ are used anyway.

The two closed-loop systems are then given by:

$$\boldsymbol{M}\dot{\boldsymbol{e}}_\nu + (\boldsymbol{D_L} + \boldsymbol{K_D})\boldsymbol{e}_\nu + \boldsymbol{K_P}\boldsymbol{e}_\eta = 0 \tag{2.18a}$$

$$\boldsymbol{M}\dot{\boldsymbol{e}}_\nu + (\boldsymbol{D_Q}(\boldsymbol{\nu}_d)\boldsymbol{\nu}_d - \boldsymbol{D_Q}(\boldsymbol{\nu})\boldsymbol{\nu}) + (\boldsymbol{D_L} + \boldsymbol{K_D})\boldsymbol{e}_\nu + \boldsymbol{K_P}\boldsymbol{e}_\eta = 0 \tag{2.18b}$$

Because of the slightly different constructions of the two control laws, the second closed-loop system (2.18b) shows an additional quadratic velocity error term. Both control laws are exponentially convergent in both velocities and positions. Due to the system simplification, the behavioral differences are very small in this example, but a small improvement on the convergence time can be observed with the second control law.

Another similar comparison is made in [118]. In this work, the system model is complete, and two control laws are produced. The first one is a completely exactly feedback-linearizing controller while the second one uses feedforward terms in the damping term. The two control laws are given by:

$$\boldsymbol{\tau} = \boldsymbol{M}(\dot{\boldsymbol{\nu}}_d + \boldsymbol{K_D}\boldsymbol{e}_\nu + \boldsymbol{K_P}\boldsymbol{e}_\eta) + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{2.19a}$$

$$\boldsymbol{\tau} = \boldsymbol{M}(\dot{\boldsymbol{\nu}}_d + \boldsymbol{K_D}\boldsymbol{e}_\nu + \boldsymbol{K_P}\boldsymbol{e}_\eta) + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu}_d)\boldsymbol{\nu}_d + \boldsymbol{g}(\boldsymbol{\eta}) \tag{2.19b}$$

Note that, in these two control laws, the PD controller terms are used as part of the reference acceleration term instead of being linearly added to the right hand side of the controller. This method essentially allows for simplifying the mass matrix in the closed-loop system equations, making them independent from the mass matrix and avoids in-

troducing nonlinear error terms in the closed-loop system. The two closed-loop systems are given as:

$$\dot{\boldsymbol{e}}_\nu + \boldsymbol{K_D}\boldsymbol{e}_\nu + \boldsymbol{K_P}\boldsymbol{e}_\eta = 0 \tag{2.20a}$$

$$\dot{\boldsymbol{e}}_\nu + (\boldsymbol{K_D} + \boldsymbol{D}(\boldsymbol{\nu}))\boldsymbol{e}_\nu + \boldsymbol{K_P}\boldsymbol{e}_\eta = 0 \tag{2.20b}$$

The closed-loop equations (2.20) are hard to interpret in terms of stability because they mix error signals in the inertial frame and in the body-fixed frame $\boldsymbol{e}_\eta$ and $\boldsymbol{e}_\nu$ respectively. A simplified approach is to consider $\boldsymbol{K_P}\boldsymbol{e}_\eta$ as a proportional term and the two other terms are derivative of the first and second order.

In this work, the addition of damping terms in the second closed-loop system leads to better trajectory tracking performances. It appears that the nonlinear damping term of this second solution behaves favorably and enhances the performances.

Of course, more combinations of feedforward and feedback terms could be used but are not discussed here. Nonetheless, as shown through all these examples, the knowledge of the model can be used to simplify the nonlinear system, lead to a linear or partially linear closed-loop system and therefore allow using a conventional tuning method for the PID like pole placement or LQR [1]. Of course, similarities with the methods presented in this section will be found in the following sections because these linearization techniques are also used with other classes of controllers.

Indeed, the model-based linearization methods introduced in this section require a precise estimation of the model parameters to perform at best. Adaptive methods introduced in section 2.5 on the other hand only require minimal model knowledge and are shown to perform as well as the model-based ones.

2.2.3 Application to Underactuated Vehicles

In the underactuated case, the linearization and PID control examples introduced in the above are not sufficient. Indeed, using the control methods introduced in section 2.2.2, some DOF would be neglected because of underactuation. The controllers of section 2.2.2 are all diagonal. They create a one-to-one relation between one output and the corresponding input. In order to take the non-actuated degrees of freedom into account, the examples introduced in this section use either a non-diagonal space

1. For more details about LQR control in this context, see [171]

reduction or a non-diagonal, kinematic couplings-based gain matrix in the control law. In both cases, the kinematic couplings of the model are used to establish a model-based kinematic guidance in a similar fashion as LOS guidance. The guidance thus created appears in the outer loop of the cascaded system. Through kinematic couplings, rotational speeds are calculated in the control law to compensate for the lack of actuation on a non-actuated translation. Other model-based methods involving less obvious coupling effects can also be found for instance in [131].

The first mechanism used in addition to model-based linearization and PID control is *asymmetrical space reduction*. Space reduction is a method consisting in reducing the spatial dimension of the system and considering the controlled or actuated DOF of the system. It is common practice when it comes to dealing with underactuated systems and especially for a system where one or more DOF are both non-actuated and naturally mechanically stable. As an example, it is common to neglect the roll motion of a torpedo shape vehicle if the restoring moment in roll is considered strong enough to keep an almost zero-roll angle during all the application time or if this DOF has no meaningful impact on the mission. In such a case, the problem can be reduced to a five-DOF problem. However, when it comes to vehicles and applications whose DOF do not *match* one-to-one, space reduction gets more complicated but offers new possibilities. In the following examples, space reduction can be considered asymmetrical because the method considerstwo sets of different DOF, one set of controlled DOF expressed at the tracking point (see section 1.1.2.1) of the system and one set of actuated DOF expressed at the center of the vehicle. Using a tracking point different from the center of the craft allows creating non-diagonal compensation behavior between the two sets of DOF.

As a first example, Ref. [3] proposes a solution to the 3-DOF position tracking problem applied to a generic uqr-craft in which space reduction is used to introduce the guidance mechanism in the control law using a virtual reference point.

In [3], a reduced version of the kinematic model equation (1.1) is produced with reduced matrices and vectors. Because $\boldsymbol{J}(\boldsymbol{\eta_2})$ and $\dot{\boldsymbol{\eta}}_{\boldsymbol{E}}$ are expressed at point E, they are reduced following the DOF required in the application: the three last rows and columns are discarded keeping only the rows corresponding to positions. On the other hand, $\boldsymbol{\nu}$ is expressed in point O_B and is therefore reduced following the actuated DOF of the ship: the second, third and fifth rows are discarded. In addition, because the transformation matrix $\boldsymbol{T}^{-1}$ is used to move from E to O_B, the rows are reduced following the DOF required in the application while the columns are reduced following the actuated DOF.

Therefore, the three last rows and columns 2, 3 and 5 are discarded in $\boldsymbol{T}^{-1}$. The reduced kinematic model is then given as:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{E},\boldsymbol{r}} = \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} \tag{2.21a}$$

$$\boldsymbol{J}(\boldsymbol{\eta})_r^{-1} = \boldsymbol{J}_{\mathbf{1}}(\boldsymbol{\eta}) \tag{2.21b}$$

$$\boldsymbol{\nu}_{\boldsymbol{E},\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{\eta}_{\mathbf{2}})_r^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E},\boldsymbol{r}} = \begin{bmatrix} u_E \\ v_E \\ w_E \end{bmatrix} \tag{2.21c}$$

$$\boldsymbol{\nu}_{\boldsymbol{r}} = \boldsymbol{T}_r^{-1}\boldsymbol{\nu}_{\boldsymbol{E},\boldsymbol{r}} = \begin{bmatrix} u \\ q \\ r \end{bmatrix} \tag{2.21d}$$

$$\boldsymbol{T}_r^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & \varepsilon_x \\ 0 & -\varepsilon_x & 0 \end{bmatrix} \tag{2.21e}$$

The reduction of the dynamic equation is more straightforward since all the matrices and vectors are expressed in point O_B and reduced in the same way, keeping the first, fifth and sixth rows and columns. In this example, $\boldsymbol{T}_r^{-1}$ behaves like a non-diagonal gain matrix creating the compensation behavior relying on the kinematic coupling of the model.

The control law presented in [3] is composed of two stages, kinematic and dynamic ones. The kinematic stage is a proportional controller with an anticipation term based on the position error calculated in $\boldsymbol{R}_{\mathbf{0}}$. The equation of the kinematic stage is given in

$\boldsymbol{R_B}$ by:

$$\boldsymbol{\nu_c} = \boldsymbol{T}^{-1}\bar{\boldsymbol{\nu}}_{\boldsymbol{E}} \tag{2.22a}$$

$$\bar{\boldsymbol{\nu}}_{\boldsymbol{E}} = \boldsymbol{J}(\boldsymbol{\eta})^{-1}\left(\dot{\boldsymbol{\eta}}^{*} + \boldsymbol{\Lambda}_{\eta}(\boldsymbol{\eta}^{*} - \boldsymbol{\eta_E})\right) - \boldsymbol{\delta}(\boldsymbol{\nu}) \tag{2.22b}$$

with $\boldsymbol{\Lambda}_{\eta}$ a definite positive gain matrix and $\boldsymbol{\delta}$ a drift vector accounting for the neglected motions and current velocity. Note that $\bar{\boldsymbol{\nu}}_{\boldsymbol{E}}$ stands for a control velocity at the tracking point E supposed to ensure convergence of the position of the tracking point towards the trajectory. All the vectors and matrices of Equation (2.22) are reduced following the steps presented before, but the index r is omitted for clarity.

The dynamic stage has a feedback linearizing structure and is calculated in the reduced space at point $\boldsymbol{O_B}$:

$$\boldsymbol{\tau_c} = \boldsymbol{M}(\dot{\boldsymbol{\nu}}_{\boldsymbol{c}} + \boldsymbol{\Lambda}_{\boldsymbol{\nu}}(\boldsymbol{\nu_c} - \boldsymbol{\nu})) + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta_2}) + \boldsymbol{d}(\boldsymbol{\nu}) \tag{2.23}$$

where $\boldsymbol{\nu_c}$ is the output velocity vector of the kinematic stage, and $\boldsymbol{d}(\boldsymbol{\nu})$ is a vector containing some terms that were discarded during the space reduction and considered as external disturbances. Equation (2.23) is expressed following the space reduction presented before.

Therefore, looking at (2.21) and (2.22), it appears clearly that the pitch and yaw control speeds at the output of the kinematic stage q_c and r_c are functions of the sway and heave control speeds in point E, $\bar{v}_E$ and $\bar{w}_E$, respectively. The pitch and yaw components of $\boldsymbol{\tau_c}$ are themselves calculated out of the lateral and vertical motions required in point E. This behavior is created by the asymmetrical space reduction and notably the introduction of non-diagonal terms in the reduced transformation matrix ${\boldsymbol{T_r}}^{-1}$.

The method developed in [3] has been extended to different propulsive topologies and applications in [168, 31]. In [168], the space reduction method is applied to the two vehicles introduced in section 1.2.1 and section 1.2.2. Both are evaluated on a four DOF task. The first vehicle is ill-actuated with respect to the task and a non-diagonal compensation mechanism is introduced with the asymmetrical space reduction of the $\boldsymbol{T_r}$ matrix. The same method is used on the second vehicle but this time neglecting the angular constraint of the task.

The space reductions in [168] are different from the one introduced in [3]. For the first vehicle, the vector and matrices expressed in O_B are reduced discarding the second and fifth rows and columns because sway and pitch are not actuated, whereas

the vectors and matrices expressed in E are reduced discarding the fourth and fifth rows and columns because roll and pitch are neglected in the task. For the second vehicle, the vectors and matrices in E are reduced keeping only the three first rows and columns corresponding to the three required translations while those expressed in O_B are reduced keeping the first, fourth and fifth rows and columns because the vehicle is actuated in surge, pitch and yaw. The shape of the vectors and matrices for both vehicles are described in detail in [168].

Partial convergence of the method is demonstrated in [31] for the first 4-DOF vehicle. The compensation mechanism allows sway tracking in E but at the cost of yaw. Nonetheless, the heading of the vehicle is kept stable thanks to hydrodynamic restoring moments and stays very close to the task requirements.

Overall, in the examples presented above, the reduced translation matrix $\boldsymbol{T}_r^{-1}$ behaves like a non-diagonal gain matrix making the velocity command of one DOF in O_B depending on the speed command on another DOF in E. As the previous solutions shown in this work, this method allows compensation of the lack of actuation over one degree of freedom with another. Preferably, a rotational DOF would be used to compensate the lack of a translation. However, one of the major issues of such methods is that reduced matrices lose some of their properties. Notably, the reduction of the $\boldsymbol{J}(\boldsymbol{\eta_2})$ might, in some cases, add new singularities to the system, making the matrix noninvertible for some orientations.

Similar behavior can be obtained without space reduction introducing a model-based non-diagonal gain matrix in the kinematic stage of the controller as displayed in [36] (only in French for now). In this example, the so-called *Handy* $\mathcal{H}$ matrix is introduced in the kinematic stage of the control law to allow compensation of the non-actuated sway motion with yaw. This matrix is autonomously calculated by an algorithm provided in this work and is based on the kinematic couplings of the model. It allows for generating the yaw moment necessary to exactly create the sway speed in the tracking point E required to cancel the lateral error. The interest of this method is that it does not require reducing the space of the application which makes generalization easy. This method is presented in details and applied to several study cases in chapter 3 and section 3.5.2.

2.3 Differential Flatness

This section introduces examples of control laws based on *differential flatness* and applied to both fully actuated and underactuated marine craft.

Differential flatness or, simply, flatness is an approach of control-command born in 1991, as a consequence of the work of four French researchers—M. Fliess, J. Lévine, Ph. Martin & P. Rouchon—on the control of an overhead crane [56]. This invention, created from an application, gave rise to a new theory of nonlinear control. It was first formulated in the language of differential algebra [55, 53] [2]. The definition of this property has been reformulated some years later in the formalism of the differential geometry of prolongations and infinite jets [52]. Flatness is a structural property of a to-be-controlled system, which naturally leads to trajectory tracking control. In the linear framework, a flat system is exactly a controllable one, so flatness is a good candidate to be a definition of controllability of nonlinear systems. Flatness is presented in detail in several books [83, 154, 116, 136, 142]. The control methodology derived from flatness proceeds in two steps: Firstly, generation of a nominal command from the to-be-followed trajectory by the system, secondly, closing the loop to have robustness properties. The closed loop can be realized in several ways: it can be classical PID, state feedback, LQG/LQR, sliding modes, model-free control... Flatness is sometimes been confused with feedback linearization, partly because, besides its definition, the notion of endogenous feedback and a related notion of linearization were introduced. Remember that flatness corresponds to the notion of state feedback linearization in the case of single-input systems. See [51] for more details on the links between flatness and feedback linearization. About ten years after the appearance of flatness, it has been demonstrated that, during operation, the trajectories of the flat output correspond to that of a linear system in Brunovský form. This property is called *exact feedforward linearization* (thus cannot be confused with feedback linearization). Through this new approach of flatness, it was possible to establish the properties, already observed in practice on applications, of robustness towards parameter errors and perturbations. We can quote [81, 80, 82] for the theoretical aspects and [39] for the practical aspects of exact feedforward linearization based on differential flatness. The flatness has spread in many domains of control; we can quote here some of them: control of mechanical systems [13], mobile robotics [54], control of electric motors [29], control of chemical reactors [140, 138,

2. This article is cited more than 3800 times at the date of writing the present work.

86] (see references in [83, 154, 116, 142] for more more applications). Besides the academic aspects, the flatness is present in many industrial realizations, too numerous to be all quoted here. More details on the application of flatness to marine craft can be found in chapter 4. The flatness prroperty of a system can also be used in the trajectory generation phase to ensure that the planned paths are adapted to the capability of the vehicle [170].

2.3.1 Application to Fully actuated vehicles

One of the few applications of differential flatness theory in the context of marine vehicles can be found in [137]. In this example, a fully actuated AUV is evaluated on a 6-DOF task. For fully actuated system, choosing the flat output is pretty straightforward. Nonetheless, this work shows that the model of the fully actuated AUV is flat for a flat output chosen as the position and orientation vector $\boldsymbol{z} = \boldsymbol{\eta}$ in the inertial frame. Then, using flatness equations, the inputs of the system are written as functions of the flat output. These expressions are then used to derive a flatness-based linearizing control law using exact feedforward linearization and PD controllers. This work also introduces an additional Kalman disturbances compensation method. The good performances of the method on a 6-DOF task are displayed in section 4.5.1.

The proof of flatness of the completely actuated AUV given in [137] is a little bit tricky and gives many details that are out the scope of the present work. Let us give a direct proof: the position and orientation vector $\boldsymbol{\eta}$ is the most obvious and natural choice for the *flat output candidate*. Choosing $\boldsymbol{z} = \boldsymbol{\eta}$ notably allows for defining the task in terms of desired position, orientation and velocities in the inertial frame, which is both logical and practical in most applications. As in many cases, the flat output found here has a strong signification w.r.t. the control problem as one wants to control $\boldsymbol{\eta}$. Then, showing the differential flatness of the system is pretty straightforward. In fact, the only variables to express as functions of the flat output are the velocity vector in the body-fixed frame $\boldsymbol{\nu}$ and the propulsion vector $\boldsymbol{\tau}$. The inverse kinematic model and the dynamic model expressed in the inertial frame constitute self-explanatory demonstrations of the flatness of the system:

$$\boldsymbol{\nu} = \boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}} \tag{2.24a}$$

$$\boldsymbol{\tau} = \boldsymbol{M}\left(\boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}} + \dot{\boldsymbol{J}}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}}\right) + \boldsymbol{C}(\dot{\boldsymbol{\eta}})\boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}} + \boldsymbol{D}(\dot{\boldsymbol{\eta}})\boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{2.24b}$$

Looking at Equation (2.24), it appears clearly that both the velocity vector $\boldsymbol{\nu}$ and the effort vector $\boldsymbol{\tau}$ can be expressed as functions of the flat output candidate $\boldsymbol{\eta}$ and its derivatives $\dot{\boldsymbol{\eta}}$ and $\ddot{\boldsymbol{\eta}}$. The model used to represent fully actuated marine craft is therefore flat with flat output $\boldsymbol{z} = \boldsymbol{\eta}$.

As a consequence of flatness, the nominal open-loop control that achieves trajectory tracking of the reference trajectory of the flat output $\boldsymbol{\eta}^*$ is expressed by:

$$\boldsymbol{\nu}^* = \boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^* \tag{2.25a}$$

$$\boldsymbol{\tau}^* = \boldsymbol{M}\left(\boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\ddot{\boldsymbol{\eta}}^* + \dot{\boldsymbol{J}}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^*\right) + \boldsymbol{C}(\dot{\boldsymbol{\eta}}^*)\boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^* + \boldsymbol{D}(\dot{\boldsymbol{\eta}}^*)\boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^* + \boldsymbol{g}(\boldsymbol{\eta}^*) \tag{2.25b}$$

Following the reasoning introduced in [81] and using the flatness equation (2.24), a flatness-based closed-loop controller can be easily derived:

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2 \tag{2.26a}$$

$$\boldsymbol{\tau}_1 = \boldsymbol{M}\boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\left(\ddot{\boldsymbol{\eta}}^* + \boldsymbol{\Lambda}(\boldsymbol{e}_\eta)\right) \tag{2.26b}$$

$$\boldsymbol{\tau}_2 = \boldsymbol{M}\dot{\boldsymbol{J}}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^* + \boldsymbol{C}(\dot{\boldsymbol{\eta}}^*)\boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^* + \boldsymbol{D}(\dot{\boldsymbol{\eta}}^*)\boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^* + \boldsymbol{g}(\boldsymbol{\eta}^*) \tag{2.26c}$$

where $\boldsymbol{\Lambda}$ is a control function like a PID controller based on the state error $\boldsymbol{e}_\eta$ and $\boldsymbol{\tau}_2$ constitutes the exact feedforward linearizing term. Of course, in the fully actuated case, the formulation of the control law built with differential flatness is very close to some of the feedforward linearizing controllers introduced in Section 2.2.

2.3.2 Application to Underactuated Vehicles

For underactuated vehicles, showing differential flatness is significantly harder. In the marine context, two very promising examples can be found in [154]. Note that, the flat output must be of the same dimension as the input. In fact, the very last example of chapter 12 in [154] states that the model of an underactuated surface ship is not differentially flat. It is impossible in this example to express all the problem variables as functions of the flat output —chosen as the positions in the horizontal plane $\boldsymbol{z} = [x\ y]$ —without expressing differential equationd. Instead, the underactuated surface ship is said to be *Liouvillian*. More details about Liouvillian systems and their properties can be found in [53, 28].

However, another very interesting system can be found in [154] as well as in [135]:

the *Hovercraft* system. Arguably, the model used to represent the hovercraft system is very close to the model of an underactuated surface vessel and could even be considered as a special case of the more generic model of the surface ship with selected numerical values of the parameters. The main differences to find between these two models are in the mass distribution and damping approximations. However, the model of the hovercraft is shown to be flat with the flat output $\boldsymbol{z} = [x\ y]$ in [154, 135]. In both examples, equations of flatness allow for calculating surge and yaw controls as function of the desired state, its derivatives up to the fourth order, and PD controllers on the flat output. More details on flatness-based control and applications can be found in chapter 4.

2.4 Sliding Mode Control

After the famous book by I. Flügge-Lotz [59] on discontinuous control, the sliding regime control was essentially introduced by a few authors in the end of the 1950s [41, 2, 60, 75, 148]. These works were followed by those of Cypkin [35], Emelyanov [42] and Itkis [93]. Utkin introduced a notion, new for the time, of sliding control applied to mono-variable classical linear systems, by the use of discontinuous controls [166]. See also [165] for a survey. All of these ideas would not have been possible without the much more theoretical work of the Soviet mathematician Filippov in the 1960s concerning differential equations with a discontinuous right-hand side [45]. The books of Utkin [166] and Sira-Ramírez [153] give a good overview of this approach to discontinuous control which has gained popularity its simplicity and its applications in various fields of automation. Moreover, these techniques have led to industrial applications. The applications of SMC in robotics and AUV start with the works of J.J.E. Slotine: [176, 155, 157, 175].

This section introduces the use of Sliding Mode Control for both fully actuated and underactuated marine craft. SMC is widely used in both linear and nonlinear systems experiencing model uncertainties or external disturbances. SMC is known for its robustness [175] and for offering theoretical finite time convergence even in the presence of model approximations and external disturbances. Finite time convergence is made possible by the introduction of a *sign* function in the controller. The use of the discontinuous signum function in SMC induces a new phenomenon called *chattering*. Chattering is a consequence of a discontinuity in the controller around the equilibrium point creating fast steep oscillations potentially damaging for the actuators. Several methods intro-

duced in this work allow mitigating or canceling chattering but at the cost of asymptotic convergence instead of finite time convergence. Many examples of successful application of SMC on surface vehicles and underwater craft can be found in the literature. This section presents some of them after briefly introducing the method on a generic simple example. This example is also used for introduction of the notations relative to SMC. The interested reader is referred to [156, 109, 117, 178] for more information about SMC outside of the marine environment. Note that, when SMC is applied to nonlinear systems feedback linearizing terms naturally appear in the controller thanks to the construction method of SMC. The end result is very similar to some PID-based controllers of section 2.2.

An example of basic SMC can be found in [156]. Let us recall the system used in the first example of Section 2.2, in the case of a n-order system, as:

$$x^{(n)} = f(\boldsymbol{X}) + g(\boldsymbol{X})u \tag{2.27}$$

The state of the system is defined as $\boldsymbol{X} = \begin{bmatrix} x & \dot{x} & \dots & x^{(n-1)} \end{bmatrix}$. Referring to the model introduced in chapter 1, matrix $g(\boldsymbol{X})$ can be seen as the inverse of a mass matrix and vector $f(\boldsymbol{X})$ as the vector regrouping all other forces and moments, notably the Coriolis and centripetal effects as well as damping. An additional vector of disturbances could be added in the system, but it is neglected in this example for the sake of simplicity. Finally, u is the system input. In this example, $f(\boldsymbol{X})$ and $g(\boldsymbol{X})$ are considered known but further analysis in the case of approximated model matrices can be found in [175, 156].

The idea behind SMC is to reduce the control problem to the lower-order problem of minimizing the distance between a point in state-space and a surface. The said *sliding surface* is in fact a state-space hypersurface[3] representing the desired dynamics of the system. It appears in the literature that a confusion is often made between the actual sliding surface and the distance between the current state and the surface. When referring to the "sliding surface", authors often use the equation of distance between the state of the system and the surface itself expressed in state space. In fact, the sliding surface itself is the set of desired states represented in state space, and it can be defined by the ensemble of system states where the distance to the surface is zero. In this work, the distance representation will often be used keeping in mind that the sliding surface

3. Recall that in a space of dimension N, a *hypersurface* is a geometric object of dimension $N-1$.

is, in fact, properly defined as the states where this distance is zero. This choice is the most common in the literature because the distance to the surface is a function of the state error which is used to close the loop in the controllers.

As will be seen later, the question of defining the sliding surface has been widely studied in the literature, but a basic definition of such a surface Σ associated with the distance measure σ is given by:

$$\sigma(\boldsymbol{X}, t) = (\frac{d}{dt} + \lambda)^{n-1} e = 0 \tag{2.28}$$

To match with the model used in this work, the example is presented with $n = 2$: $\sigma(\boldsymbol{X}, t) = \dot{e} + \lambda e$. The quantity λ is a design parameter representing the slope of the surface in state space. It can be tuned for better performances or relatively to the application. Equation (2.28) shows that, in trajectory tracking applications, the sliding surface Σ moves with the reference. One can also observe that the equilibrium point $\dot{e} = 0$, $e = 0$ is contained in the surface $\sigma = 0$. Therefore, the control problem becomes a problem of minimization of the distance σ. Once the surface is reached, the *sliding motion* drives the system on the surface towards the equilibrium point $\dot{e} = 0$, $e = 0$ giving its name to the method.

The design of the command vector $u(t)$ relatively to the sliding surface is based on the Lyapunov theory. Let us define the following Lyapunov function candidate:

$$V = \frac{1}{2}\sigma^2 \tag{2.29}$$

for which one obviously has $V(0) = 0$ and $V(\sigma) > 0$ for $\sigma \neq 0$. The first order time derivative of V is:

$$\dot{V} = \dot{\sigma}\sigma \tag{2.30}$$

and the stability criteria $\dot{V} < 0$ therefore leads to:

$$\dot{\sigma}\sigma < 0 \tag{2.31}$$

Equation (2.31) is referred to as the *sliding condition*. The sliding condition must be respected to ensure stability. The evolution of the sliding surface is then chosen to meet the sliding condition at all time. One basic solution is:

$$\dot{\sigma} = -\gamma \operatorname{sign}(\sigma) \tag{2.32}$$

where γ is a control gain to be tuned later. Using $\dot{\sigma} = \ddot{e} + \lambda\dot{e}$, the model Equation (2.27) can be combined to the sliding condition (2.32) to give an expression of the control vector u:

$$u = g^{-1}(\boldsymbol{X})[\ddot{x}^* - \lambda\dot{e} - \gamma\,\mathrm{sign}(\sigma)] - f(\boldsymbol{X}) \tag{2.33}$$

Equation (2.33) can also be written as:

$$u = u_1 + u_2 \tag{2.34a}$$
$$u_1 = g^{-1}(\boldsymbol{X})[\ddot{x}^* - \lambda\dot{e}] - f(\boldsymbol{X}) \tag{2.34b}$$
$$u_2 = -g^{-1}(\boldsymbol{X})\gamma\,\mathrm{sign}(\sigma) \tag{2.34c}$$

where the so-called *equivalent term* u_1 is in fact a feedback linearizing term of reference $\ddot{x}^*$ (see Section 2.2), and u_2 is the switching term assuring convergence towards the sliding surface. In the case of an approximated model, u_1 would be based on the model matrices approximation and would only compensate the known parts of the model matrices.

Remark 2.1 *The conventional sliding surface definition used in this example and in most of the following may lead to the appearance of steady state error in the presence of external unknown disturbances. This phenomena is described in appendix A as well as an alternative sliding surface definition called Integral SMC. The sliding surfaces used in most of the following examples can be replaced with integral sliding mode surfaces allowing better robustness to unknown external disturbances. The conventional sliding surface presented here has a proportional derivative structure which can lead to steady state error and notably when using smooth switching functions. Adding an integral term in the sliding surface definition creates a PID behavior.*

Note that, in a real application, the simple controller (2.33) is likely to create chattering as the switching of the signum function would not be instantaneous. The following examples propose alternative switching functions reducing or avoiding chattering. Figure 2.3 gives an overview of some of the switching functions used in the following examples to avoid chattering. This functions are continuous around the equilibrium point which limits the fast oscillations causing chattering but also modify the convergence of the controllers. Some work has been done to compare different switching strategies and [134] shows that SM is globally exponentially stable with this selection of smooth

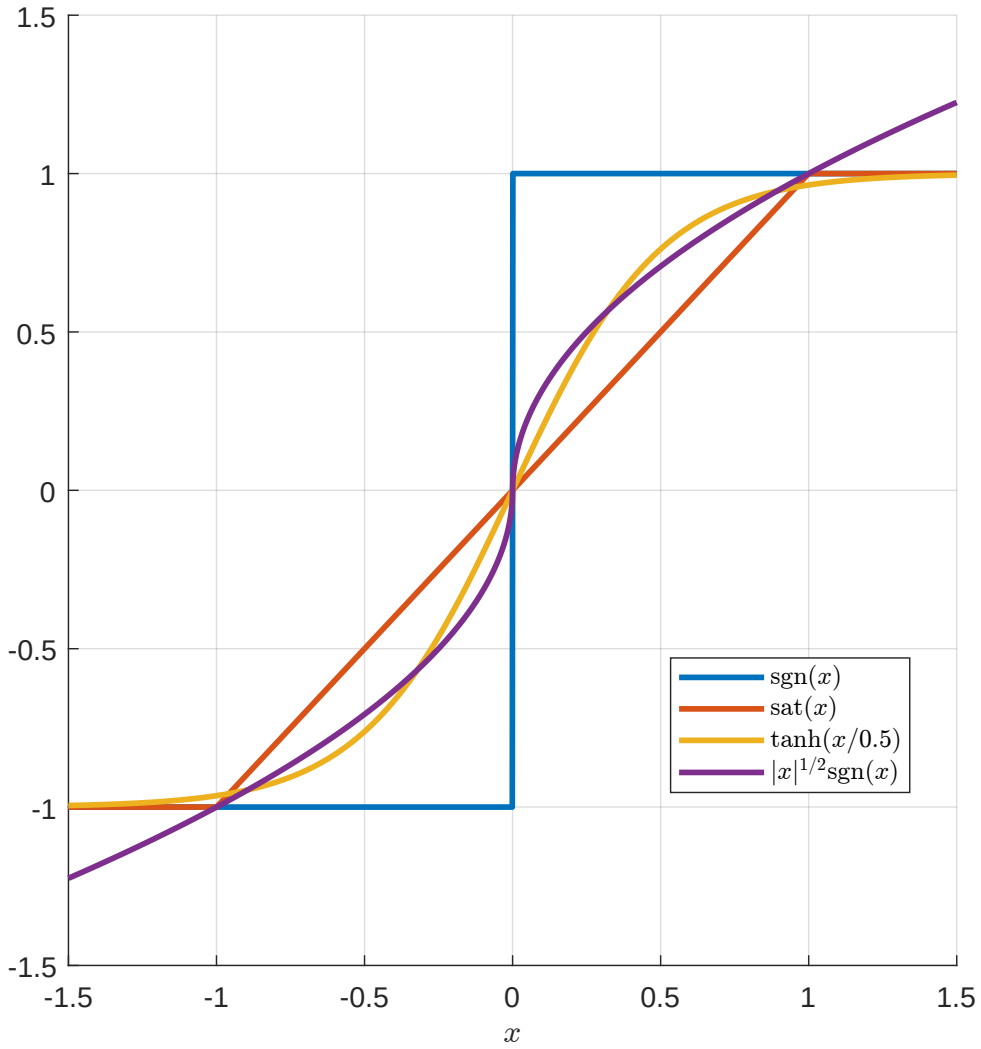


Figure 2.3 – Comparison of the switching functions used to avoid chattering.

switching functions.

It is also worth mentioning the works of [147, 78, 159, 96] where SMC is combined with Fuzzy logic to create more robust and adaptive controllers. Adaptive sliding mode controllers can also be found in the work of [151, 150].

2.4.1 Application to Fully Actuated Vehicles

The work of [175] gives a good example of SMC applied to fully actuated marine craft, and the formulation is the same as in the example exposed before. The system has three actuated DOF and is sent on a horizontal plane application composed of (x, y, ψ)

trajectories. In [175], the model is not entirely known a priori and estimated values of the model matrices are used in the controller. The model also includes approximated disturbances. All approximations are bounded, and this work shows that, when they are known, the estimation bounds can be used for the calculation of the gain parameter γ (noted $K(X,t)$ in the reference) to ensure convergence. In addition, Ref. [175] displays the use of a different switching function based on a *saturation* function instead of a signun function. The saturation function shown in figure 2.3 is better suited for real applications since it avoids the chattering phenomenon. Using the saturation function allows for creating a boundary layer around the sliding surface where the switching effect is made continuous. The size of the boundary layer can be tuned for adapted behavior of the system. The saturation function is originally defined as:

$$\text{sat}(x) = \begin{cases} \text{sign}(x) & \text{if } |x| >= 1 \\ x & \text{if } |x| < 1 \end{cases}$$

Using the saturation function, it appears clearly that, inside the boundary layer $|\sigma| < 1$, the nonlinear controller (2.33) becomes very similar to the PID-based linearizing controllers introduced in Section 2.2. Inside of the boundary layer, the controller becomes:

$$u = g^{-1}(\boldsymbol{X})[\ddot{x}^* - \lambda\dot{e} - \gamma\sigma] - f(\boldsymbol{X}) \tag{2.35}$$

Therefore, the main drawback of using a continuous switching function such as the saturation function is that the convergence cannot be guarantied in finite time anymore. Nonetheless, [175] gives a method for calculation of the control gain, the slope of the surface and the boundary layer thickness based on the estimation of the model matrices, and shows very good tracking results.

The method is then used for the design of three decoupled controllers, one per actuated DOF. For each DOF, a sliding surface is defined as well as the different control parameters such as the boundary layer thickness or the switch control gain. The three sliding surfaces are given as:

$$\sigma_u = e_u + \lambda_u e_x \tag{2.36a}$$

$$\sigma_v = e_v + \lambda_v e_y \tag{2.36b}$$

$$\sigma_r = e_r + \lambda_r e_\psi \tag{2.36c}$$

The surge and sway surfaces given by (2.36a) and (2.36b), respectively, are defined in terms of positions in the inertial frame $\boldsymbol{R_0}$ and velocities in the body-fixed frame $\boldsymbol{R_B}$. While this could be fine for simple mono-axial applications, such a definition of the surge and sway sliding surfaces may be problematic when it comes to more complex trajectories. In this case, the trajectory tracking results show good performances on the tested trajectories.

The work of [34] is another good example of SMC following the same overall logic. Here, the vehicle is controlled only in the diving plane and is fully actuated w.r.t. the task. In this work, the performances of four different controllers based on SMC are compared. The main difference between the four controllers is the model used in the equivalent part U_1 (see below Equation (2.39)). The first one uses a linearized model, the second one uses the nonlinear exact model, the third one uses an adaptive model, and the last one uses estimated states. The compliance and robustness of the SMC method make such differences between controllers applied to the same system possible and is once again demonstrated by the results of this work.

The definition of the sliding surface is also slightly different in [34] since a first order linear surface, defined by $\sigma = \boldsymbol{\gamma}^\top \boldsymbol{x}$, is used. Here, $\boldsymbol{x} \in \mathbb{R}^3$ is a reduction of the state of the vehicle in the diving plane, $\boldsymbol{\gamma} \in \mathbb{R}^3$ is a vector of gains, and σ contains, in fact, three decoupled surfaces, one per DOF, linearly added. The use of first order sliding surfaces allows for a simpler command vector and enables the use of pole-placement techniques when tuning the parameters.

Another different definition of the set of sliding surfaces is given in [87]. As in [175], the sliding surfaces are defined in terms of the position errors in the inertial frame summed to velocity errors in the moving frame. The set of sliding surfaces is given by:

$$\boldsymbol{\sigma}(\boldsymbol{e_\nu}, \boldsymbol{e_\eta}) = \begin{bmatrix} \boldsymbol{\Lambda_1} & \boldsymbol{\Lambda_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{e_\nu} \\ \\ \boldsymbol{e_\eta} \end{bmatrix} \tag{2.37}$$

where $\boldsymbol{\Lambda_1}$ and $\boldsymbol{\Lambda_2}$ are coefficient matrices defined in $\mathbb{R}^{6\times6}$ and identified in the following. To take better account of the coupling effects between the DOF of the system, the sliding surfaces used here are defined over the full state space while, in the references presented before, they are often defined only on the output. For ease of understanding of the following equations, another self-explanatory expression of the vehicle model is given

to match both the notation of chapter 1 and the notation used in the reference:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} = \boldsymbol{f}(\boldsymbol{\nu}, \boldsymbol{\eta}) + \boldsymbol{g}(\boldsymbol{\nu}, \boldsymbol{\eta})\boldsymbol{u}(t) \tag{2.38a}$$

$$\dot{\boldsymbol{\eta}} = \boldsymbol{h}(\boldsymbol{\nu}, \boldsymbol{\eta}) \tag{2.38b}$$

The definition of the sliding surfaces (2.37) leads to a different expression of the control vector:

$$\boldsymbol{U} = \boldsymbol{U_1} + \boldsymbol{U_2} + \boldsymbol{U_3} \tag{2.39a}$$

$$\boldsymbol{U_1} = \hat{\boldsymbol{g}}(\boldsymbol{\nu}, \boldsymbol{\eta})^{-1} \left[(\boldsymbol{\Lambda_1 M}^{-1})^{-1} \dot{\boldsymbol{\nu}}^* - \hat{\boldsymbol{f}}(\boldsymbol{\nu}, \boldsymbol{\eta}) \right] \tag{2.39b}$$

$$\boldsymbol{U_2} = \hat{\boldsymbol{g}}(\boldsymbol{\nu}, \boldsymbol{\eta})^{-1} (\boldsymbol{\Lambda_1 M}^{-1})^{-1} \boldsymbol{\Lambda_2} \left[\dot{\boldsymbol{\eta}}^* - \hat{\boldsymbol{h}}(\boldsymbol{\nu}, \boldsymbol{\eta}) \right] \tag{2.39c}$$

$$\boldsymbol{U_3} = -\hat{\boldsymbol{g}}(\boldsymbol{\nu}, \boldsymbol{\eta})^{-1} (\boldsymbol{\Lambda_1 M}^{-1})^{-1} \boldsymbol{F}(\boldsymbol{\sigma}, \Phi) \tag{2.39d}$$

In (2.39), the estimates $\hat{\boldsymbol{f}}$, $\hat{\boldsymbol{g}}$ and $\hat{\boldsymbol{h}}$ of the model function are used. A similar definition of the control vector $\boldsymbol{U}$ could be given with the real values of the model functions if they were to be considered known. As for the usual definition of SMC given earlier in this section, the control vector $\boldsymbol{U}$ can be broken down into three parts. $\boldsymbol{U_1}$ compensates the estimates of the dynamic effects of the model, $\boldsymbol{U_2}$ provides stabilization based on estimates of the positional elements in $\boldsymbol{h}$, and $\boldsymbol{U_3}$ is the switching term driving the system to the sliding surfaces. This work uses the hyperbolic tangent function over the boundary layer of thickness Φ as a switching function:

$$\boldsymbol{F}(\boldsymbol{\sigma}, \Phi) = \boldsymbol{\gamma} \tanh(\frac{\boldsymbol{\sigma}}{\Phi}) \tag{2.40}$$

The hyperbolic tangent function is displayed in figure 2.3 with $\Phi = 0.5$. The hyperbolic tangent appears ideal since it is smooth around the equilibrium point $\boldsymbol{\sigma} = 0$ but is still steep enough around the sliding surface to ensure fast switching and the sliding behavior.

Here, again, the structure of the control law is very similar to some controllers introduced in Section 2.2 and notably when the distance to the sliding surface is close to zero.

As for the saturation function seen before, using the hyperbolic tangent avoids chattering at the cost of a non-finite convergence time.

In order to set the values of $\boldsymbol{\Lambda_1}$ and $\boldsymbol{\Lambda_2}$, the system is linearized. Doing so highlights that the first parameter matrix $\boldsymbol{\Lambda_1}$ can be chosen as the identity matrix without loss of generality in the fully actuated case. Then, analyzing the linearized close-loop dynamics leads to a *desirable* choice of the second parameter matrix $\boldsymbol{\Lambda_2}$:

$$\boldsymbol{\nu} = \boldsymbol{\Lambda_2}\boldsymbol{\eta} \tag{2.41}$$

Equation (2.41) is quite similar to the kinematic model of the system given in equation (1.1).

The authors then apply the SMC method they introduced on a set of four Single-Input-Multiple-States subsystems. The three main subsystems are tested first independently then all together in association with LOS guidance in a waypoint tracking experiment. Effects of sea current are also highlighted in the last simulations.

Over the years, some more work has been carried out on SMC and notably the introduction of new sliding surfaces. As an example, *terminal sliding mode* introduced in [117] and applied to the marine craft model in [113] or [112] is based on a different sliding surface definition. Terminal sliding mode offers finite convergence time, faster and more precisely than conventional sliding mode by introduction of a new nonlinear term in the sliding surface definition. Note that, in this work, the sliding surfaces are completely defined in the inertial frame. In fact, two formulations of the terminal sliding surface are given in [113]:

$$\boldsymbol{\sigma_c} = \dot{\boldsymbol{e}}(t) + \boldsymbol{\lambda}\boldsymbol{e}(t)^{\frac{q}{p}} = 0 \tag{2.42a}$$

$$\boldsymbol{\sigma_n} = \boldsymbol{e}(\boldsymbol{t}) + \frac{1}{\boldsymbol{\lambda}}\dot{\boldsymbol{e}}(t)^{\frac{p}{q}} = 0 \tag{2.42b}$$

In (2.42), λ is analogous to the control parameter used in the previous examples, and p and q are two positive odd integers satisfying $p > q$. Using the sliding condition (2.31), two control vectors can be derived from (2.42), respectively:

$$\boldsymbol{\tau_\eta} = \hat{\boldsymbol{M}}_{\boldsymbol{\eta}}\left(\ddot{\boldsymbol{\eta}}^* + \boldsymbol{\lambda}\frac{q}{p}\boldsymbol{e}(t)^{\frac{q}{p}-1}\dot{\boldsymbol{e}}(t) + \boldsymbol{\gamma}\,\mathrm{sign}(\boldsymbol{\sigma_c})\right) + \hat{\boldsymbol{N}}_{\boldsymbol{\eta}}(\boldsymbol{\nu}, \boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \tag{2.43a}$$

$$\boldsymbol{\tau_\eta} = \hat{\boldsymbol{M}}_{\boldsymbol{\eta}}\left(\ddot{\boldsymbol{\eta}}^* + \boldsymbol{\lambda}\frac{q}{p}\dot{\boldsymbol{e}}(t)^{2-\frac{p}{q}} + \boldsymbol{\gamma}\,\mathrm{sign}(\boldsymbol{\sigma_n})\right) + \hat{\boldsymbol{N}}_{\boldsymbol{\eta}}(\boldsymbol{\nu}, \boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \tag{2.43b}$$

The first expression of the terminal sliding surface (2.42a) shows a singularity in $\boldsymbol{e}(t) =$

0 because $\frac{q}{p} < 1$. Therefore, the second expression (2.42b) is to be used. Of course, the two expressions of (2.42) are equivalent when the system reaches the surface. Once again, the signum function is substituted with a saturation function in the final controllers.

Simulation results show the performances of the terminal sliding mode in comparison with traditional SMC and a classical computed torque controller (CTC). Terminal sliding mode seems to outperform the traditional SMC and CTC notably displaying better convergence times and a smoother overall behavior on the helix tracking task.

Another example of a different formulation of the sliding surface can be found in [4]. This work proposes a controller based on *Super Twisting* Sliding Mode Control (STSMC) applied to the linearized model of the vehicle in the diving plane. More details about the theory behind STSMC, sliding order and sliding accuracy can be found in [109, 95]. STSMC is designed to have as good disturbance rejection and robustness as traditional SMC but reducing the chattering effect without the need of substituting the sign function, therefore assuring finite time convergence. The Super Twisting behavior is created using a different solution of the sliding condition (2.31):

$$\dot{\sigma} = \gamma_1 |\sigma|^{\frac{1}{2}} \operatorname{sign}(\sigma) + \gamma_2 \int_0^t \operatorname{sign}(\sigma(\zeta)) d\zeta \tag{2.44}$$

The continuous switching behavior is created by the first member of equation (2.44) displayed on figure 2.3. The second member of the sliding surface definition creates an integral effect on the sliding surface canceling possible steady state errors.

Using the small angles approximation to linearize the system in the diving plane, one can derive the sliding condition given by (2.44) into the following control law:

$$u = \frac{1}{d_1}(-c_{11}w - c_{12}q - c_{13}\theta - c_{14}z + u_0\dot{\theta} - \lambda\dot{e_z} + \gamma_1 \sigma^{\frac{1}{2}} \operatorname{sign}(\sigma) + \gamma_2 \int_0^t \operatorname{sign}(\sigma(\zeta)) d\zeta) \tag{2.45}$$

In this case, a constant depth z^*, $\dot{z^*} = \ddot{z^*} = 0$ is tracked. The c_{ij} coefficients are model parameters and the control input u is analogous to the stern plan defection angle δ_s. u_0 is a constant sway speed.

The simulation results indeed show dampening of the chattering effect when the STSMC is compared with traditional SMC. However, it should be noted that STSMC

displays slightly slower convergence than SMC.

This section demonstrated the use of different sliding mode controls in the case of fully actuated marine vehicles. As seen in this section, sliding mode control comes in different shapes. However, the two main levers for tweaking the controllers are the definition of the sliding surface itself and the choice of the solution of the sliding condition. Many more examples would be worth adding to this section like the work in [179] where SMC is associated with LOS guidance and *Fuzzy Logic* in the diving control of an autonomous bio-mimetic dolphin robot as well as [149], which introduces four decoupled SM controllers of different orders in a vehicle with actuated surge, heave, pitch and yaw.

2.4.2 Application to Underactuated Vehicles

This section studies the use of SMC in the case of underactuated marine craft. It is notably interested in how the SMC method can be used as a guidance principle and allow compensation of underactuation in the same way LOS does and how SMC can be coupled with external guidance laws.

As a first example, in [10], the model of a ur-ship tracking horizontal straight lines is studied. Two decoupled controllers are designed one for the surge motion and the other for the yaw motion. Two sliding surfaces are given:

$$\sigma_u = e_u + \lambda_u \int_0^t e_u(\zeta) d\zeta \tag{2.46a}$$

$$\sigma_r = \dot{e}_v + 2\lambda_r e_v + \lambda_r^2 \int_0^t e_v(\zeta) d\zeta \tag{2.46b}$$

There are multiple points of interest in (2.46). First, both sliding surfaces are completely defined in the mobile frame. However, because positions in the mobile frame can not be measured, the surfaces are defined in terms of the integral of the velocity error on the two axes of the moving frame. This solution is analogous to defining the surfaces in terms of positions [4] but, as shown later, the definition of the surge and sway references matters when it come to tracking trajectories originally defined in the inertial frame. Then, while the surge sliding surface σ_u is of the first order relatively to the integral of e_u, the yaw sliding surface σ_r is of the second order and defined relatively

4. Note that the definition of the sliding surfaces (2.46) is not considered as *Integral* SMC as seen in appendix A

to the sway speed error e_v instead of the heading angle or the yaw velocity. Therefore, the yaw controller derived from σ_r will be based on the sway speed error e_v. Doing so allows the authors to compensate the lack of sway actuation with yaw. This method is very close to the other guidance methods exposed earlier in this review, and it is somewhat equivalent to generating yaw controls based on the lateral error measured on the y_B axis of the moving frame.

The surge and yaw controllers are derived from the respective sliding surfaces and the dynamic model of the system as:

$$\tau_u = -m_{22} v r + d_1 u - m_{11}(-\dot{u}^* + \lambda_u e_u) - \gamma_u \operatorname{sign}(\sigma_u) \tag{2.47a}$$

$$\tau_r = \frac{m_{33}}{u^* - \frac{m_{11}}{m_{22}} u} \left[\frac{r \tau_u}{m_{22}} - v_r - v_{rd} - (2\lambda_r \dot{e}_v + \lambda_r^2 e_v) - \gamma_r \operatorname{sign}(\sigma_r) \right] \tag{2.47b}$$

Note that, in the original article, estimated values of the model parameters are used in the control calculations, and the authors left the possibility to consider nonlinear damping. Equation (2.47) has been simplified for clarity. Equation (2.47b) is of a different structure than the other control equations seen in this section because it is based on the first time derivation of the sway equation of the dynamic model instead of the first order derivative of the chosen sliding condition solution (2.46b). Hence the new speed terms v_r and v_{rd} which contain the terms created by this manipulation. Exact definitions of v_r, v_{rd} as well as the gains γ_u and γ_r can be found in [10]. The dynamic sway equation derivative is used instead of the sliding condition derivative because the latter would require information upon the second order derivative of both the actual and desired sway speeds. Such information is not available in this work.

However, as claimed in [181, 40], the solution proposed in [10] does not solve all the tracking problems. In fact, defining the sliding surfaces in terms of the integrals of the velocity errors in the moving frame leads to neglecting constant offsets of the desired position signals. To counteract this problem, the work of [181] and [40] redefine the surge and sway velocity references.

First [181], gives a similar approach for trajectory tracking of a ur-ship in the horizontal plane but with modified velocity references. To take a possible position offset into account, the desired reference surge and sway speeds are calculated as:

$$u^* = \cos\psi \dot{x}^* + \sin\psi \dot{y}^* - k\cos\psi e_x - k\sin\psi e_y \tag{2.48a}$$

$$v^* = -\sin\psi \dot{x}^* + \cos\psi \dot{y}^* + k\sin\psi e_x - k\cos\psi e_y \tag{2.48b}$$

with k a positive constant control parameter. Note that, in the original work [181], two different control parameters k_1 and k_2, are mentioned but do not appear in the equations. This work shows that convergence of the surge and sway errors e_u and e_v leads to convergence of the position errors in the inertial frame e_x and e_y. The first order derivative of the second order yaw sliding surface expressed in terms of the integral of the sway error is used for the calculation of the yaw control. Therefore, knowledge about the second order derivative of the desired sway motion is necessary as well as the third order derivative of both desired position signals.

In the same way, [40] proposes a more global solution than in [10], solving the case of offset trajectories. To do so, new surge and sway speed references are defined as:

$$\begin{bmatrix} u^* \\ v^* \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \dot{x}^* + l_x \tanh(-\frac{k_x}{l_x} e_x) \\ \dot{y}^* + l_y \tanh(-\frac{k_y}{l_y} e_y) \end{bmatrix} \tag{2.49}$$

where k_x and k_y are controller gains, and l_x and l_y are saturation coefficients chosen relatively to the system's physics. The definition of the speed references u^* and v^* given by Equation (2.49) leads to the error equation:

$$\begin{bmatrix} e_u \\ e_v \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \dot{e}_x + l_x \tanh(-\frac{k_x}{l_x} e_x) \\ \dot{e}_x + l_y \tanh(-\frac{k_y}{l_y} e_y) \end{bmatrix} \tag{2.50}$$

Equation (2.50) shows that convergence of the speed errors e_u and e_v to zero leads to asymptotic convergence of the position errors e_x and e_y. In fact, the speed errors in Equation (2.50) behave nearly like sliding surfaces for the positions. The rotation matrix being non-singular, when the speed errors converge to $e_u = e_v = 0$, one obtains:

$$\dot{e}_x - l_x \tanh(-\frac{k_x}{l_x} e_x) = 0 \tag{2.51a}$$

$$\dot{e}_y - l_y \tanh(-\frac{k_y}{l_y} e_y) = 0 \tag{2.51b}$$

Equation (2.51) can be seen as the sliding conditions of the inner loop of the system assuring asymptotic convergence of the position errors the hyperbolic tangent function being used instead of the signum function. In a way, this controller is quite similar to the two-staged controllers presented in section 2.2.

The sliding surfaces used for the dynamic controllers are then given. The sliding surfaces used in [40] are very similar to (2.46) but with two different control parameters in the yaw surface:

$$\sigma_u = e_u + \lambda_u \int_0^t e_u(\zeta) d\zeta \tag{2.52a}$$

$$\sigma_r = \dot{e_v} + \lambda_{r,1} e_v + \lambda_{r,2} \int_0^t e_v(\zeta) d\zeta \tag{2.52b}$$

Here, too, the sliding surface σ_r used for yaw control is built upon the sway error e_v. Slightly different dynamics than the usual sliding condition (2.31) are imposed on σ_u and σ_r:

$$\dot{\sigma}_u = -\gamma_{u,1}\sigma_u - \gamma_{u,2}\,\mathrm{sign}(\sigma_u) \tag{2.53a}$$

$$\dot{\sigma}_v = -\gamma_{v,1}\sigma_v - \gamma_{v,2}\,\mathrm{sign}(\sigma_v) \tag{2.53b}$$

where $\gamma_{i,j}$ are all strictly positive control parameters. Note that using this sliding condition adds new terms, equivalent to proportional and derivative terms, in the controllers.

This time, the surge and yaw controllers are both calculated with the first order derivations of the sliding surfaces giving:

$$\tau_u = -X_u u - a_{23} v r + \frac{1}{M_{11}} \left(\dot{u}^* - \lambda_u e_u - \gamma_{u,1}\sigma_u - \gamma_{u,2}\,\mathrm{sign}(\sigma_u) \right) \tag{2.54a}$$

$$\begin{aligned} \tau_r = &-N_r r - a_{12} u v + \frac{1}{b}(-M_2(Y_v \dot{v} + a_{13}\dot{u} r)\Gamma - \lambda_{r,1}\dot{e}_v - \lambda_{r,2} e_v \\ &- \gamma_{r,1}\sigma_r - \gamma_{r,2}\,\mathrm{sign}(\sigma_r)) \end{aligned} \tag{2.54b}$$

The newly introduced mass coefficients a_{12}, a_{23}, M_{11} and M_2 as well as control parameters Γ and b are given in detail in [40]. The damping surge, sway and yaw coefficients X_u, Y_v and N_r, respectively, are issued from the damping matrix of the dynamic model. The simulation results show good performance in tracking linear trajectories with or without an offset and circular trajectories.

A more recent example of SMC applied to underactuated vehicles can be found in [172]. In this work, the authors propose a Super Twisting SMC based solution for the problem of leader–follower tracking. The followers are uqr-vehicles following the trajectory set by the mother ship. Three dimensions LOS guidance is used to calculate

the approach angle references in pitch and yaw. Then, three decoupled Super Twisting sliding mode controllers are designed for surge, pitch and yaw as well as a fourth one looping back on the forward speed of the mother ship.

For surge control of the follower submarines, a zero-order sliding surface is built upon the surge error with a constant reference u^*. The surface is therefore given as $\sigma_u = e_u$ and the Super Twisting formulation of the sliding condition is used as in [4]

$$\dot{\sigma}_u = -\gamma_{u,1}|\sigma_u|^{\frac{1}{2}}\,\mathrm{sign}(\sigma_u) - \gamma_{u,2}\int_0^t \mathrm{sign}(\sigma_u(\zeta))d\zeta \tag{2.55}$$

with two strictly positive gain parameters $\gamma_{u,1}$ and $\gamma_{u,2}$. The definition of the kinematic Super Twisting sliding-mode controller calculating the surge speed consign of the mother ship is similar to this one. The mother ship velocity is calculated with the position error kinematic model. More details about mother ship control can be found in the reference [172].

On the other hand, the pitch and yaw controllers are Super Twisting sliding mode controllers using first order terminal sliding surfaces as in [113]. The pitch and yaw sliding surfaces and associated sliding conditions are given by

$$\sigma_\theta = e_\theta + \lambda_\theta|\dot{e_\theta}|^{\frac{p_\theta}{q_\theta}} \tag{2.56a}$$

$$\dot{\sigma}_\theta = -\gamma_{\theta,1}|\sigma_\theta|^{\frac{1}{2}}\,\mathrm{sign}(\sigma_\theta) - \gamma_{\theta,2}\int_0^t \mathrm{sign}(\sigma_\theta(\zeta))d\zeta \tag{2.56b}$$

$$\sigma_\psi = e_\psi + \lambda_\psi|\dot{e_\psi}|^{\frac{p_\psi}{q_\psi}} \tag{2.56c}$$

$$\dot{\sigma}_\psi = -\gamma_{\psi,1}|\sigma_\psi|^{\frac{1}{2}}\,\mathrm{sign}(\sigma_\psi) - \gamma_{\psi,2}\int_0^t \mathrm{sign}(\sigma_\psi(\zeta))d\zeta \tag{2.56d}$$

where all the control parameters are constant and strictly positive.

As seen in the previous examples, the actual control signals τ_u, τ_q and τ_r are derived from the first order derivative of the associated sliding surface and calculated with the corresponding dynamic model equations. For increased robustness to external disturbances, adaptive disturbance terms can be added to the controller equations as shown in [172]. Finally, the four sliding mode controllers are shown to stabilize the tracking errors of the following submarines.

These few examples show that SMC can be used in the underactuated case and displays good performances. In these examples, one can find the idea of using a rotational degree of freedom to compensate the lack of one or two non actuated translations.

Sliding mode itself can be used as guidance principle like in [10, 40, 181] where yaw controls are given as functions of the lateral motion errors using the same idea as in the LOS examples given earlier [106] directly in the sliding surface calculations.

2.5 Adaptive Control

While this work is essentially focused on model-based control methods, adaptive and model-free controllers are worth mentioning. These methods have received a great amount of interest in the past few years in and outside the scope of marine vehicles [97]. With this type of controllers, the model of the vehicle and the disturbance to which it is subjected to are considered unknown or partially known. These controllers adapt during the application either by changing the gain parameters or estimating the external effects on the vehicle. Adaptive controllers are therefore naturally robust to external disturbance and environment changes.

One of the first example of adaptive controllers applied to underwater vehicles can be found in [177] where an adaptive sliding mode controller is experimented on an underwater vehicle. Later, [6] proposes several adaptive controllers to compensate the effects of an unknown sea current using either adaptive gain parameters or a current velocity estimator. Following the same idea, [183] proposes a disturbance observer associated with a self-adjusting controller with adaptive gains. This work also raises that chattering can occur when using self-adjusting control gains and proposes an alternative formulation dampening the chattering effects. The interested reader is also referred to [9, 71, 118, 158, 182], for examples of PID-based controllers using adaptive structures for linearization. Other examples of adaptive sliding-mode controllers can be found for instance in [76].

In [141], two adaptive controllers are presented. The first one is based on a delayed estimate of the model parameters. The second one uses the known bounds of the nonlinear effects of the model to estimate and compensate the *worst-case scenario* of the disturbing effects. This second controller is shown to be much more efficient than the first one.

Several other methods have been developed over the years to adjust the gains of an adaptive controller or estimate the disturbances such as in [114] which uses fuzzy logic to compute gains or [185] which is based on neural network estimation of the state.

Arguably, integral action is the first step to adaptability in the controllers and is

sufficient to compensate most of the effects acting on an AUV [7]. This integral action can take place at the guidance level to create new equilibriums and compensate, for example, sea current as shown in [16, 24]. In the case of adaptive LOS guidance, [61] shows that the integral term can be used in several places to compensate the crab angle for instance.

One type of adaptive controller is described in more details in the next Section as it used in the following developments, the so-called *Intelligent PID* (iPID) controller.

2.5.1 Intelligent PID

Model-free control based on iPIDs is described in details in [49] which unifies the previous works done on the subjected in the references therein. It is a wide spread model-free control method (more than 400 citations to date) applied to a large range of systems from shape memory alloy actuators [79] to flying drones [33, 11], AUVs [15], autonomous land vehicles [123] and some more industry and research applications.

The idea behind iPIDs is to approximate the system with a very simple linear *ultra-local* model where all the unknown effects are summed up in a single estimator function. It is therefore easily adapted to any system, the ultra-local model and estimation technique are almost always the same (the order of the ultra-local model may change) [50]. The disturbing term of the ultra-local model is estimated with an *ALIEN* filter. Some of the early work done on ALIEN filter can be found for instane in [120] where they are used for estimation of noisy signals.

2.5.1.1 Ultra-local model

The perhaps complex model of the system is estimated using the following ultra-local model:

$$y^{(\nu)} = \alpha u + F \tag{2.57}$$

where y is the system output, u is the input, ν is the order of the ultra-local model (may differ from the order of the system itself), α is a non-physical constant tuning parameter and F is the continuously updated function estimating all the unknown effects (nonlinear model effects, disturbances).

In most cases, a first order ultra-local model $\nu = 1$ gives good results but sometimes a second order model is preferable. Note that several solutions can exist to approximate a system with an ultra-local model. As an example, consider the second order model

representing the dynamic of the position of a vehicle. It is possible to use a second order ultra-local model using the position of the vehicle as a measurement but it is also possible to use a first order ultra-local model using the speed of the vehicle as the measurement. Finally, it could also be possible to use a first order ultra-local model with position measurement and consider all second order effects as disturbance.

It is straightforward to build a controller based on the ultra-local model (2.57). In this example, a simple proportional control law with a feedforward term is used but more advanced iPID controllers can be useful.

The controller based on the first order ultra-local model is:

$$u = \frac{1}{\alpha}\left(\dot{y}^* - \hat{F} + K_p e\right) \tag{2.58}$$

where $\dot{y}^*$ is the first order derivative of the desired output y^*, $e = y^* - y$ is the error between the desired and current output and K_p is a gain parameter. $\hat{F}$ is a short term estimate of F.

The closed-loop temporal is then:

$$\dot{e} + K_p e = \hat{F} - F \tag{2.59}$$

Stability of the closed-loop system (2.59) is hardly demonstrated because of the behavior of the estimate $\hat{F}$ but the large number of successful applications of the method tends to serve as a proof.

The work of [48] shows that iPs and iPIs do not work for second order ultra-local models. In these cases, a derivative term is needed. Also, the work of [127] shows that the estimation of F in iPs and iPDs has an integral effect and makes iPs (respectively iPDs) equivalent to conventional PIs (respectively PIDs). Consequently, iPDs are favord in the following.

2.5.1.2 Estimation of F

Looking at equation (2.58), it appears clearly that the main difficulty of iPIDs is the estimation of F. The estimation of F is performed using *ALIEN* filters introduced in [58]. The adaptive term F regroups all the unmodeled effects whether they are nonlinear effects of the system or external disturbance with no distinction. The calculations presented in this section are for a first order ultra-local model. Same methodology can

be applied to higher orders.

The first hypothesis to estimating $\hat{F}$ is that it is piece-wise constant on a small time window $[t_{k-1}, t_k]$. The ultra-local model at the first order should therefore be expressed as:

$$\dot{y}(t) = \alpha u(t) + F_k, t \in [t_{k-1}, t_k[\tag{2.60}$$

In the following, the subscript k will be omitted for simplicity and the calculation are performed on the time window $[0, T]$ with $0 < T \ll 1$.

Equation (2.60) gives:

$$F = \dot{y} - \alpha u \tag{2.61}$$

Because F is considered constant on the time window, equation (2.61) is given in operational form:

$$\frac{F}{s} = sy(s) - y(0) - \alpha u(s) \tag{2.62}$$

The initial condition $y(0)$ is eliminated differientiating once with respect to s:

$$-\frac{F}{s^2} = y(s) + s\frac{dy}{ds}(s) - \alpha\frac{du}{ds}(s) \tag{2.63}$$

To add a filtering effect, equation (2.63) is multiplied by $1/s^2$ so that every member is integrated at least once when going back to temporal space. Doing so adds robustness to the estimation. Equation (2.63) becomes:

$$-\frac{F}{s^4} = \frac{1}{s^2}y(s) + \frac{1}{s}\frac{dy}{ds}(s) - \alpha\frac{1}{s^2}\frac{du}{ds}(s) \tag{2.64}$$

Back to temporal, equation (2.64) becomes:

$$-\frac{T^3}{3!}F = \int_0^T \left(\frac{(T-\tau)^1}{1!} - \tau\right) y(\tau)d\tau - \alpha\int_0^T \frac{(T-\tau)^1}{1!}(-\tau)u(\tau)d\tau \tag{2.65}$$

which simplifies as:

$$F = -\frac{6}{T^3}\int_0^T \left[(T-2\tau)y(\tau) + \alpha(T-\tau)\tau u(\tau)\right] d\tau \tag{2.66}$$

The estimate can then be shifted into the time window $[t-T, t]$:

$$F = -\frac{6}{T^3}\int_0^T \left[(T-2\tau)y(\tau+t-T) + \alpha(T-\tau)\tau u(\tau+t-T)\right] d\tau \tag{2.67}$$

The estimate of F (2.67) is adjusted at each sample and used in the controller to compensate all the nonlinear effects and disturbances. No strict rules are given for the tuning of parameter α except that it should be chosen so that the input and output of the system are of the same order of magnitude. The works of [127] and [133] give guidelines to the tuning of the control parameter based on trial-error.

2.6 Conclusions on the literature review

This section shows some of the tremendous work done on the control of autonomous marine craft. A first clear conclusion is that there are many different methods in the literature answering the problem of controlling a marine vehicle with seemingly equivalent performances. It is then a matter of adapting the choice of the control method to the application, vehicle, sensors, etc... The four main solutions studied in this section — Line Of Sight guidance, PID-based control and linearization, Differential flatness and Sliding Mode Control — all show very promising results. These methods are used and detailed further on in the following. The choice of the control method may also be related to the type of scenario at hand in the task. Notably, LOS guidance is mainly used in setpoint regulation and path following or path tracking applications while differential flatness is more suited for trajectory tracking scenarios.

Also, this section shows that the control of underactuated vehicle requires additional mechanisms to quit the traditional one-to-one input-output relationship of conventional controllers and create necessary compensation behaviors. Often times, the idea with underactuated vehicle is to use one of the actuated DOF to compensate the lack of another non-actuated DOF. The classical example of this strategy is the traditional *ur*-vessel where yaw is used to cope for the lack of sway. This strategy can be extended in 3D space and with other associations of actuated rotations and non-actuated translations. The two novel control results presented in chapter 3 and chapter 4 follow this same logic.

CHAPTER 3

A NEW KINEMATIC GUIDANCE PRINCIPLE BASED ON THE MATRIX $\mathcal{H}$

Contents

This chapter introduces a new model-based kinematic guidance principle for underactuated marine vehicles based on the new *Handy* matrix $\mathcal{H}$. This controller is notably

useful for trajectory tracking applications of ill-actuated AUVs. It relies on the idea of compensating the lack of actuation of one translation with an actuated rotation. This idea appears indirectly in many of the controllers presented for underactuated vehicles in chapter 2. The $\mathcal{H}$ matrix acts as a non-diagonal gain matrix based on model coefficients. The compensation mechanism created by the matrix $\mathcal{H}$ is in fact a kinematic guidance principle included in the control law. Same principle appears for instance in the SMC for underactuated AUVs presented in section 2.4.2.

Outside of the scope of marine vehicles, other model-based methods are used for compensation of ill-actuation of mechanical systems. As an example, the *partial feedback linearization* introduced in [161, 160] relies on dynamic coupling effects between active and passive DOF. Under some restrictions, this work shows that passive DOF can be controlled exploiting their relations with active ones. Note that some marine crafts meet the *strong inertial coupling* condition introduced in [161] and could potentially be controlled using the *non-collocated* controller.

The work of Spong on *collocated* and *non-collocated* control as well as the multiple articles of Fossen on LOS guidance are behind the idea of the control method introduced in this chapter. The $\mathcal{H}$ matrix and the associated controller have been designed to allow exploiting the natural kinematic couplings of the vehicle to control some of the passive DOF using the actuated DOF.

3.1 A theoretical example of model based compensation mechanisms

In this section, as a theoretical example, a linear system is studied to demonstrate the methodology used in the design of the $\mathcal{H}$ matrix controller. First, an intuitive model-based controller is designed for the fully-actuated system. This controller is based on the inverse model of the system and a Proportional-Integral control law. It gives straightforward independent convergence of every state components of the system towards the desired value, naturally canceling any coupling terms of the system. Then, an underactuated version of the system is introduced and the first controller is shown to be insufficient for proper control of the underactuated system. Finally, the model-based compensation mechanism is introduced in the controller by means of a new model-based asymmetrical term and is shown to allow convergence of the system towards the

trajectory.

3.1.1 Intuitive controller in the fully actuated case

Before introducing the $\mathcal{H}$ matrix controller in the context of an underwater vehicle, the method is described on a simple, theoretical example. As it will be demonstrated later on, this example mimics the kinematic model of an AUV restricted to the horizontal plane. Let us introduce the following fully-actuated system:

$$\dot{x}_1 = u_1 \tag{3.1a}$$
$$\dot{x}_2 = u_2 + \varepsilon u_3 \tag{3.1b}$$
$$\dot{x}_3 = u_3 \tag{3.1c}$$

where the input vector is $\boldsymbol{u} = [u_1 \ u_2 \ u_3]^\top$ and the state is $\boldsymbol{x} = [x_1 \ x_2 \ x_3]^\top$. The input-output relationship of system (3.1) can be rewritten as $\dot{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{u}$ with:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \varepsilon \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

In this example, the system is fully actuated and the off-diagonal term of $\boldsymbol{T}$ creates a coupling effect between the third input component u_3 and the second DOF of the state x_2 equation (3.1b). As demonstrated later, this example is equivalent to controlling the velocity of a given point on a solid body $\dot{\boldsymbol{x}}$ ($\dot{x}_1$ and $\dot{x}_2$ being linear velocities and $\dot{x}_3$ being an angular speed) with the velocity of another point $\boldsymbol{u}$ (u_1 and u_2 are linear velocities, u_3 is an angular speed) through the transformation matrix $\boldsymbol{T}$.

We now build a model-based controller to drive the state of the system $\boldsymbol{x}$ towards a desired state $\boldsymbol{x}^*$. In this first example, the system is fully actuated and can therefore track all three components of the desired state. The following intuitive controller can

be designed:

$$\boldsymbol{u} = \boldsymbol{T}^{-1}(\dot{\boldsymbol{x}}^* + \boldsymbol{\lambda}(\boldsymbol{e_x})) \tag{3.3a}$$

$$\boldsymbol{e_x} = \boldsymbol{x}^* - \boldsymbol{x} \tag{3.3b}$$

$$\boldsymbol{\lambda}(\boldsymbol{e_x}) = \boldsymbol{K_p}\boldsymbol{e_x} + \boldsymbol{K_i}\int_0^t \boldsymbol{e_x}(\zeta)d\zeta \tag{3.3c}$$

$$\boldsymbol{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\varepsilon \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3d}$$

The control input vector $\boldsymbol{u}$ is built using the inverse of model (3.1) where a feedforward term $\dot{\boldsymbol{x}}^*$ associated with a compensator function $\boldsymbol{\lambda}(\boldsymbol{e_x})$ have been plugged in place of the derivative of the state $\dot{\boldsymbol{x}}$. In this example, $\boldsymbol{\lambda}$ contains three decoupled PI controllers based on the tracking error $\boldsymbol{e_x}$ with $\boldsymbol{K_p}$ and $\boldsymbol{K_i}$ two appropriate diagonal matrices of control gains. Other usual controllers could be used instead.

This kind of structure for a controller is very well known and notably in the autonomous marine vehicles community (See section 2.2.2 or [63] for examples). With this controller, the closed-loop system is:

$$\dot{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{T}^{-1}(\dot{\boldsymbol{x}}^* + \boldsymbol{\lambda}(\boldsymbol{e_x})) = \dot{\boldsymbol{x}}^* + \boldsymbol{\lambda}(\boldsymbol{e_x}) \tag{3.4a}$$

$$\dot{\boldsymbol{e_x}} + \boldsymbol{K_p}\boldsymbol{e_x} + \boldsymbol{K_i}\int_0^t \boldsymbol{e_x}(\zeta)d\zeta = 0 \tag{3.4b}$$

which, for appropriate choices of the gain parameters (diagonal, positive definite), ensures global exponential convergence of the three components of the state towards the desired values.

Note that the model-based structure of the control vector $\boldsymbol{u}$ naturally cancels the coupling effects of the non-diagonal matrix $\boldsymbol{T}$ and allows independent convergence of the two DOF x_2 and x_3.

3.1.2 Kinematic guidance in the underactuated case

Consider now that the second input component u_2 of the previous system (3.1) is not available anymore, the control vector is now of form $\boldsymbol{u} = [u_1 \ 0 \ u_3]^\top$. The system

is now considered underactuated with respect to the task $\boldsymbol{x}^* = [x_1^* \; x_2^* \; x_3^*]^\top$. The task is then reduced to the first two components x_1^* and x_2^* to match the new capabilities of the system. Yet, because the actuated DOF of the system do not exactly match the required DOF of the task, it is ill-actuated with respect to the reduced task and a compensating behavior will be mandatory.

Very often, when working with an underactuated system, the non-actuated DOF are neglected and the controller is built on a reduced version of the state space [3, 31]. In this example, the reduced version of equation (3.3) is:

$$\begin{bmatrix} u_1 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ \dot{x}_3^* + \lambda_3(e_{x_3}) \end{bmatrix} \tag{3.5}$$

where the second row of the problem corresponding to the non-actuated DOF has been removed. In order to keep a square relationship, the second column of $\boldsymbol{T}^{-1}$ has also been discarded in (3.5), hence neglecting the coupling effect created by the off-diagonal term in the second row. The control functions λ_1 and λ_3 are marked differently to show that the set of parameters of each of the PI controller can be different.

The reduced controller (3.5) would perfectly work for any application where the first and third DOF x_1 and x_3 are controlled but, in this case, where x_2 is supposed to be controlled, this reduced controller is not applicable. Using this reduced controller on the system (3.1) gives the following closed-loop system:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \varepsilon \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ 0 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ \varepsilon u_3 \\ u_3 \end{bmatrix} = \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ \varepsilon(\dot{x}_3^* + \lambda_3(e_{x_3})) \\ \dot{x}_3^* + \lambda_3(e_{x_3}) \end{bmatrix} \tag{3.6}$$

Equation (3.6) indeed shows that the first and third DOF are well controlled towards the desired values x_1^* and x_3^* but the second DOF is not. As expected after space reduction, the second component of the desired state x_2^* does not even appear in the closed-loop system (3.6). However, the third input component u_3 appears to act on the second DOF x_2 though the coupling term ε. This relationship means that the third control input u_3 could be used to control either x_2 or x_3.

The idea of a model-based compensating controller is then to use this natural cou-

pling effect of the model to generate a control input u_3 allowing to track the desired value of x_2. To do so, a new non-diagonal term has to be added to the expression of the controller, the so-called *Handy Matrix* $\mathcal{H}$. The new controller $\bar{\boldsymbol{u}} = [\bar{u}_1\ 0\ \bar{u}_3]^\top$ is defined as:

$$\bar{\boldsymbol{u}} = \mathcal{H}\boldsymbol{T}^{-1}(\dot{\boldsymbol{x}}^* + \boldsymbol{\lambda}(\boldsymbol{e_x})) \tag{3.7}$$

The Handy matrix is designed to create a new expression of $\bar{u}_3$ which, when the controller is applied to the system, allows tracking of x_2. Therefore, the new control input $\bar{u}_3$ will be a function of the desired state x_2^* and of the correction term term $\lambda_2(e_{x_2})$. Input $\bar{u}_3$ must also cancel any dependence on x_3 or x_3^* in the closed-loop equation of x_2. The new controller will be calculated on the complete space and not on a reduced space.

In this first example, matrix $\mathcal{H}$ is built iteratively to demonstrate the process but an algorithm computing $\mathcal{H}$ for any applicable system is provided in the following sections. Detailing equation (3.7) gives:

$$\begin{bmatrix} \bar{u}_1 \\ 0 \\ \bar{u}_3 \end{bmatrix} = \mathcal{H} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\varepsilon \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1^* + \lambda(e_{x_1}) \\ \dot{x}_2^* + \lambda_2(e_{x_2}) \\ \dot{x}_3^* + \lambda_3(e_{x_3}) \end{bmatrix} = \mathcal{H} \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ \dot{x}_2^* + \lambda_2(e_{x_2}) - \varepsilon(\dot{x}_3^* + \lambda_3(e_{x_3})) \\ \dot{x}_3^* + \lambda_3(e_{x_3}) \end{bmatrix} \tag{3.8}$$

On the other hand, applying the control vector $\bar{\boldsymbol{u}}$ to the system (3.1) gives:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \bar{u}_1 \\ \varepsilon\bar{u}_3 \\ \bar{u}_3 \end{bmatrix} \tag{3.9}$$

The control inputs $\bar{u}_1$ and $\bar{u}_3$ must create the following expected closed-loop expressions for $\dot{x}_1$ and $\dot{x}_2$:

$$\dot{x}_1 = \dot{x}_1^* + \lambda_1(e_{x_1}) \tag{3.10a}$$
$$\dot{x}_2 = \dot{x}_2^* + \lambda_2(e_{x_2}) \tag{3.10b}$$

Considering Equations (3.8), (3.9) and (3.10), the coefficients of $\mathcal{H}$ can be reverse-

engineered; The relation $\bar{u}_1 = \dot{x}_1^* + \lambda_1(e_{x_1})$ gives the first line of $\mathcal{H}$: $\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$. The zero on the second line of $\bar{\boldsymbol{u}}$ gives the second line of $\mathcal{H}$: $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$.

For the third line, the previous discussion hints that a first term will be needed in the second column to turn $\bar{u}_3$ into a function of $\dot{x}_2^* + \lambda_2(e_{x_2})$. Another term in the third column will also be necessary to cancel the dependence in x_3^*. The third line of $\mathcal{H}$ will then be of shape $\begin{bmatrix} 0 & a & b \end{bmatrix}$ and coefficients a and b are found using Equations (3.9) and (3.10):

$$\bar{u}_3 = \frac{1}{\varepsilon}(\dot{x}_2^* + \lambda_2(e_{x_2})) = \begin{bmatrix} 0 & a & b \end{bmatrix} \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ \dot{x}_2^* + \lambda_2(e_{x_2}) - (\dot{x}_3^* + \lambda_3(e_{x_3})) \\ \dot{x}_3^* + \lambda_3(e_{x_3}) \end{bmatrix} \tag{3.11}$$

The solution of this equation is given by $a = \frac{1}{\varepsilon}$ and $b = 1$. The handy matrix $\mathcal{H}$ for this example is therefore:

$$\mathcal{H} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{\varepsilon} & 1 \end{bmatrix} \tag{3.12}$$

and the controller reads:

$$\begin{bmatrix} \bar{u}_1 \\ 0 \\ \bar{u}_3 \end{bmatrix} = \mathcal{H} \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ \dot{x}_2^* + \lambda_2(e_{x_2}) - \varepsilon(\dot{x}_3^* + \lambda_3(e_{x_3})) \\ \dot{x}_3^* + \lambda_3(e_{x_3}) \end{bmatrix} = \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ 0 \\ \frac{1}{\varepsilon}(\dot{x}_2^* + \lambda_2(e_{x_2})) \end{bmatrix} \tag{3.13}$$

Matrix $\mathcal{H}$ is non-diagonal and presents a form of symmetry with the model matrix

$\boldsymbol{T}$. The closed-loop system for this example is then:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \varepsilon \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{\varepsilon} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\varepsilon \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ \dot{x}_2^* + \lambda_2(e_{x_2}) \\ \dot{x}_3^* + \lambda_3(e_{x_3}) \end{bmatrix} = \begin{bmatrix} \dot{x}_1^* + \lambda_1(e_{x_1}) \\ \dot{x}_2^* + \lambda_2(e_{x_2}) \\ \frac{1}{\varepsilon}\left(\dot{x}_2^* + \lambda_2(e_{x_2})\right) \end{bmatrix} \tag{3.14}$$

Equation (3.14) shows that the new control vector $\bar{\boldsymbol{u}}$ of (3.7) allows tracking of the two first DOF x_1 and x_2. The third DOF x_3 is disturbed by the coupling effect. The disturbance on x_3 must be assessed to ensure that the controller will not generate problematic configuration. Ideally, x_3 should be naturally stable for better use of this controller.

In this example, the application of the non-diagonal matrix $\mathcal{H}$ to the system exploits the natural coupling of the system to create the expected compensation behavior and allows to control the non-actuated second DOF x_2 using the third input $\bar{u}_3$.

3.2 Application to the kinematic model of the underwater vehicle

The model-based compensation strategy introduced in the previous section is applied, in this section, to the kinematic model of an underactuated underwater vehicle in the form of the *Handy* matrix $\mathcal{H}$. This section shows that the kinematic model of an underwater vehicle with a off-centered tracking point is very similar to the example system of section 3.1 and can therefore be treated in the same way. Several unitary cases of compensation are given and the algorithm for calculation of the Handy matrix in the general case is detailed and applied to the example of a standard uqr-craft. This compensation mechanism will then be used as a kinematic model-based guidance principle on underactuated underwater craft.

3.2.1 Interest of a tracking point

In order to reproduce a structure similar to (3.1) in the model of the underwater craft, this method uses an off-centered tracking point as defined in section 1.1.2.1. Tracking points or *virtual reference points*(VRP) appear in the literature [14, 3] as a way to add angular stability to the vehicle. The VRP is usually positioned at the bow of the

vehicle and controlling the VRP is equivalent ot pulling the vehicle from this point. It is more naturally stable than pushing from the rear or the center, similar to how pulling a trolley is steadier than pushing it from a single rear point [14].

Another noteworthy aspect of using a tracking point is to recreate a similar structure as equation (3.1) in the kinematic model of the vehicle. Indeed, equation (1.3) shows similarities with the example system (3.1) introduced in section 3.1 with the addition of the transformation matrix $\boldsymbol{J}(\boldsymbol{\eta})$. The input of the subsystem (1.3) is $\boldsymbol{\nu}$ and the output is $\boldsymbol{\eta_E}$. The modified kinematic model is recalled here:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{T}\boldsymbol{\nu} \tag{3.15}$$

A controller relying on the model-based compensation mechanism introduced in section 3.1 can then be designed for an underactuated underwater vehicle with a tracking point $\boldsymbol{E}$. Following the theoretical example detailed above, the compensating controller applied to the kinematic model of the vehicle will be of shape:

$$\boldsymbol{\nu} = \mathcal{H}\boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\tilde{\boldsymbol{\eta}}}_{\boldsymbol{E}} \tag{3.16}$$

where $\dot{\tilde{\boldsymbol{\eta}}}_{\boldsymbol{E}}$ denotes a corrective velocity vector expressed in the inertial frame at the tracking point. This vector is a function of the desired velocity vector and of the error between the actual and desired state of the tracking point. It is typically defined as:

$$\dot{\tilde{\boldsymbol{\eta}}}_{\boldsymbol{E}} = \dot{\boldsymbol{\eta}}^{*} + \boldsymbol{\lambda}(\boldsymbol{e}_{\eta}) \tag{3.17}$$

where $\boldsymbol{e}_{\eta} = \boldsymbol{\eta}^{*} - \boldsymbol{\eta}_{\boldsymbol{E}}$ is the error vector between the desired state $\boldsymbol{\eta}^{*}$ and the state of the tracking point $\boldsymbol{\eta}_{\boldsymbol{E}}$ and $\boldsymbol{\lambda}$ is a vector of decoupled PI controllers, one per DOF.

3.2.2 Unitary cases

This Section introduces the unitary cases of compensation that can be combined to design a complete controller. The matrix $\boldsymbol{T}$ of (1.4) shows that multiple coupling effects can appear on the same DOF. Depending on the position of the tracking point $\boldsymbol{E}$ and the propulsive arrangement of the system, different compensation mechanisms can be created.

As a rule of thumb, if $\boldsymbol{E}$ has a non-zero coordinate on one axis of $\mathcal{R}_B$, say the $\boldsymbol{x_B}$

axis, then the rotation around one of the two other axis, for instance $\boldsymbol{z_B}$, can be used to generate a translation of point $\boldsymbol{E}$ on the third axis, $\boldsymbol{y_B}$ in this case.

Two binary vectors are introduced to represent the actuation of the system: $\boldsymbol{h_{O_B}}$ and $\boldsymbol{h_E}$. They respectively represent the actuated DOF of the vehicle and the DOF controlled at the tracking point $\boldsymbol{E}$ in the mobile frame. Both vectors have one column and six rows, one per DOF, where "1" means actuated or controlled and "0" means non-actuated or not controlled. As an example, a torpedo-shaped uqr-vessel (See section 5.1 for an example of such a vehicle, Remus100) with a tracking point positioned at the front and following a 3-D (x, y, z) trajectory with no attitude constraint can be represented by the following pair of vectors:

$$\boldsymbol{h_{O_B}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{h_E} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.18}$$

The vector $\boldsymbol{h_{O_B}}$ encodes that the surge, pitch and yaw of the vehicle are actuated at the center of the vehicle $\boldsymbol{O_B}$ and the vector $\boldsymbol{h_E}$ encodes that the surge, sway and heave of the tracking point E are controlled. This pair of vectors implies that pitch and yaw, both in association with surge, have been chosen to compensate for the lack of sway and heave actuation and therefore allows the vehicle to fulfill tasks consisting in three translations and no rotations.

The $\boldsymbol{h_{O_B}}$ vector of a system is imposed by the propulsive configuration whereas the $\boldsymbol{h_E}$ vector is chosen depending on the position of the tracking point and the task. Associating the controlled DOF of the tracking point $\boldsymbol{h_E}$ and the task can be tricky because of the nonlinear relationship between the translations coded in $\boldsymbol{h_E}$ and defined in $\mathcal{R}_B$ and the necessary translations of the task defined in $\mathcal{R}_0$. If the task requires two translations with no additional angular constraints then any pair of controlled translations of the tracking point can be used to fulfill the task providing that the vehicle is capable of maintaining the right attitude. As an example, if a task is composed of an horizontal trajectory with no angular constraint, the vehicle must be able to move along the $\boldsymbol{x_0}$

and $\boldsymbol{y}_o$ axes. Two orthogonal translations are needed. The obvious choice of controlled DOF for this task is surge and sway (translations on the $\boldsymbol{x_B}$ and $\boldsymbol{y_B}$ axis). But, if the vehicle is also capable of maintaining a constant $90°$ roll angle, surge and heave could be used instead. In the same way, if the vehicle can maintain a constant $90°$ pitch angle, sway and heave would also allow fulfilling the task. Therefore, the choice of $\boldsymbol{h_E}$ must take the rotation capabilities of the vehicle into account.

In order to place the tracking point for a given task and vehicle or to choose the propulsive arrangement of a vehicle for a given task and tracking point, table 3.1 gives a list of elementary compensation possibilities.

$\boldsymbol{E}$	$\boldsymbol{h_{O_B}}$	$\boldsymbol{h_E}$
$[\varepsilon_x\ 0\ 0]^\top$	$[1\ 0\ 0\ 0\ 0\ 1]^\top$	$[1\ 1\ 0\ 0\ 0\ 0]^\top$
	$[1\ 0\ 0\ 0\ 1\ 0]^\top$	$[1\ 0\ 1\ 0\ 0\ 0]^\top$
$[0\ \varepsilon_y\ 0]^\top$	$[0\ 1\ 0\ 1\ 0\ 0]^\top$	$[0\ 1\ 1\ 0\ 0\ 0]^\top$
	$[0\ 1\ 0\ 0\ 0\ 1]^\top$	$[1\ 1\ 0\ 0\ 0\ 0]^\top$
$[0\ 0\ \varepsilon_z]^\top$	$[0\ 0\ 1\ 1\ 0\ 0]^\top$	$[0\ 1\ 1\ 0\ 0\ 0]^\top$
	$[0\ 0\ 1\ 0\ 1\ 0]^\top$	$[1\ 0\ 1\ 0\ 0\ 0]^\top$

Table 3.1 – Unitary cases of compensation according to the position of the tracking point $\boldsymbol{E}$

table 3.1 shows the unitary cases of compensation of a non-actuated translation with an actuated rotation according to the position of the tracking point $\boldsymbol{E}$. It shows which DOF of the tracking point can be controlled depending on its position and the propulsive arrangement of the vehicle.

Several design rules can be drawn from this table:

1. This method can only be used if the tracking point $\boldsymbol{E}$ is different from $\boldsymbol{O_B}$ the center of $\mathcal{R}_B$.
2. This method only allows compensation of translations with rotations.
3. This method cannot be used to compensate a translation on one axis with a rotation around the same axis.

Of course, the unitary cases of table 3.1 can be combined and associated with additional actuated DOF to create a complete propulsive configuration. Note that this set of rules

can also be used to understand what kind of task a given AUV is able to complete.

Additional remarks can be made about table 3.1. First, note that, in each case, at least two DOF appear in both $\boldsymbol{h}_{O_B}$ and $\boldsymbol{h}_E$, one being an actuated translation. As an example, with the tracking point $\boldsymbol{E}$ of coordinates $[\varepsilon_x \ 0 \ 0]^\top$, yaw can be used to compensate sway or pitch can be used to compensate heave but, in both cases, actuation in surge is mandatory. From a kinematic point of view, this additional actuated speed is not strictly necessary to perform compensation of a non actuated linear speed. But, from a control perspective, this DOF is mandatory. It covers the cases where a 90 degrees rotation of the vehicle is not enough to cancel the error on the compensated linear DOF. In such a scenario, the error is transferred to this other, actuated translation and can be canceled.

Note also that each $\boldsymbol{h}_E$ vector appears twice in table 3.1. Therefore, two equivalent control solutions can be found for two different propulsive configurations and the same set of controlled DOF. As an example, a vehicle actuated in sway and roll with a tracking point on the $\boldsymbol{y}_B$ axis and a vehicle actuated in heave and roll with a tracking point on the $\boldsymbol{z}_B$ axis can both be chosen for a task requiring sway and heave.

As exposed later in this work, using the $\mathcal{H}$-based control method can give several solutions for following a task with a given vehicle. It will then be the user's decision to pick among these solutions the more suitable one. One criterion is the need of controlled rotations. Indeed, when a rotation is used for compensation it cannot be controlled anymore. As an example, a ur-surface vessel cannot use this method to control both sway and yaw at the same time. However, the uncontrolled rotation is, most of the times, mechanically stable or stabilized by hydrodynamic effects on the vehicle. In the example of the ur-vessel, yaw naturally tends towards the tangent to the track when sway error is canceled and does not need to be actively controlled.

To demonstrate the different possibilities given by this method, simulations of some of the unitary cases of table 3.1 have been conducted. These simulations aim to show that the $\mathcal{H}$-based controller can create different vehicle behaviors. It is of course a very simple elementary example and not a realistic task. The trajectory of this example is a vertical rail traveled at constant speed with no angular constraint. Considering that the vehicle arrives at the top of the rail in a neutral horizontal state (no roll or pitch but any yaw), heave control of the tracking point is mandatory. In facts, the necessity of heave actuation at the tracking point is due to the initial attitude of the vehicle and its need to instantaneously move along the $\boldsymbol{z}_0$ axis. If the vehicle had a $90°$ initial pitch angle and

were passively stable in this configuration then surge actuation would be sufficient to fulfill the task. Moreover, this application is a trajectory tracking scenario, the vehicle is supposed to track the position and velocity of the target at all time. It is not allowed any uncontrolled transitory regime as it could be the case in setpoint regulation scenarios. In such cases, the tracking performances would be degraded locally but heave actuation as such would not be necessary. This simple example also demonstrates the impact of the initial condition of a task or, more broadly, the implications of having subsequent fairly simple pieces of trajectory which, when put together, form a more complex task.

Table 3.1 gives two solutions: a tracking point on the $\boldsymbol{x_B}$ axis with surge and pitch actuation or a tracking point on the $\boldsymbol{y_B}$ axis with sway and roll actuation. In the first solution, the coordinates of the tracking point are $[\varepsilon\ 0\ 0]^\top$. The corresponding vectors of actuated and controlled DOF as well as the $\mathcal{H}$ matrix used in this case are:

$$\boldsymbol{h_{O_B}}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \boldsymbol{h_E} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathcal{H}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/\varepsilon & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

In the second solution, the coordinates of the tracking points are $[0\ \varepsilon\ 0]^\top$ and the vectors of actuated and controlled DOF as well as the $\mathcal{H}$ matrix are:

$$\boldsymbol{h_{O_B}}^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{h_E} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \mathcal{H}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\varepsilon & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

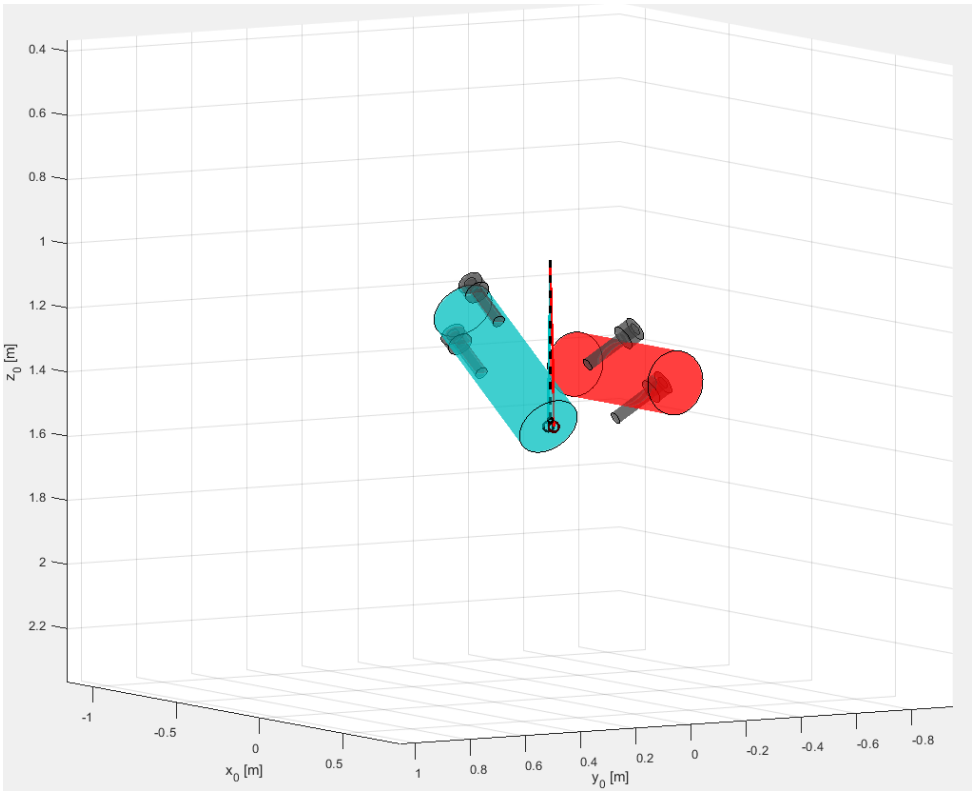


Figure 3.1 – Simulation snapshot at $t = 1s$. Blue vehicle: Pitch compensation - Red vehicle: Roll compensation

In this example, the tracking point is outside of the hull fo the vehicle to keep kinematic equivalence with the first case. It also allows showing that the tracking point does not need to be a physical point of the vehicle. Any point of $\mathcal{R}_B$can be chosen as a tracking point.

The different behaviors created by the two control strategies appear clearly on the simulation snapshot figure 3.1. On figure 3.1, each colored cylinder represents the hull of a vehicle. The blue one uses pitch compensation while the red one uses roll compensation. They are both equipped with two thrusters represented with grey cylinders. The tracked trajectory is represented with the dashed black line and the tracking point of the two vehicle are figured with a black circle. Both tracking point overlap with the current target on this snapshot showing perfect position tracking in both cases.

Figure 3.2 shows that the two compensating solutions are equivalent in terms of position tracking and notably regarding the z_o axis. On the other hand, figure 3.3 shows the difference of strategy of the two compensation solutions. The first vehicle (Blue line)

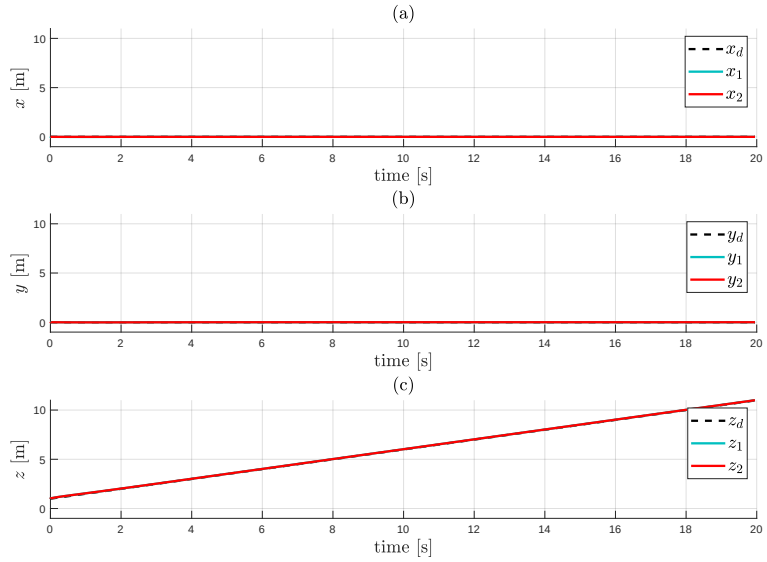


Figure 3.2 – Comparison of the position of the vehicles in the two cases. Blue: Pitch compensation - Red: Roll compensation. a: $\boldsymbol{x}_\mathbf{o}$, b: $\boldsymbol{y}_\mathbf{o}$, c: $\boldsymbol{z}_\mathbf{o}$

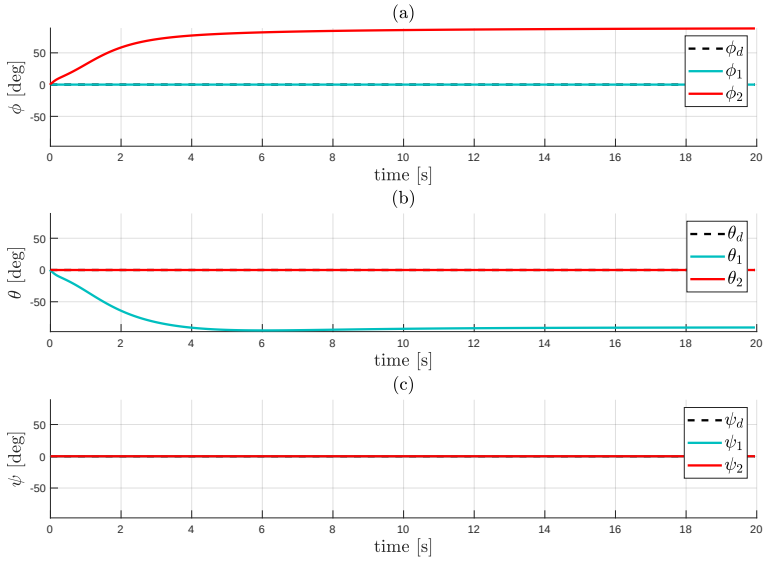


Figure 3.3 – Comparison of the orientation of the vehicles in the two cases. Desired orientation values are arbitrarily put to 0 but are not part of the task. Blue: Pitch compensation - Red: Roll compensation. a: Roll, b: Pitch, c: Yaw

pitches down to generate a heave motion at the tracking point as shown on figure 3.3.(c) whereas the second vehicle (Red Line) rolls on the side to generate heave motion as shown on figure 3.3.(a).

This elementary example illustrates the necessary choice of strategy using this method. Also, if the task also required the vehicle to keep a neutral pitch for instance, then the second solution using roll for compensation would be preferable. On the other hand if the vehicle were equipped with a camera pointing downwards then roll should not be disturbed and the first solution should be chosen. Additional constraints would limit the possible solutions. The vehicle being suitably equipped, it is also possible to alternate between two strategies during the mission, the needed $\mathcal{H}$ matrix being recalculated to meet the chosen strategy with the algorithm presented in the following.

3.2.3 Algorithm for the calculation of $\mathcal{H}$

This section details the algorithm calculating the Handy matrix $\mathcal{H}$ for a given system and task. The computation of $\mathcal{H}$ makes use of the skew-symmetrical cross-product matrix $\boldsymbol{S}(\boldsymbol{\lambda})$. Notably, for point $\boldsymbol{E}$ of coordinates $[\varepsilon_x\ \varepsilon_y\ \varepsilon_z]^\top$, it calculates $\boldsymbol{S} = \boldsymbol{S}([1/\varepsilon_x\ 1/\varepsilon_y\ 1/\varepsilon_z]^\top)$:

$$\boldsymbol{S}\left(\begin{bmatrix}1/\varepsilon_x\\1/\varepsilon_y\\1/\varepsilon_z\end{bmatrix}\right) = \begin{bmatrix}0 & -1/\varepsilon_z & 1/\varepsilon_y\\1/\varepsilon_z & 0 & -1/\varepsilon_x\\-1/\varepsilon_y & 1/\varepsilon_x & 0\end{bmatrix} \tag{3.19}$$

Note that, to avoid singularities, the matrix $\boldsymbol{S}$ is built only with the non-zero coordinates of $\boldsymbol{E}$. As an example, for a tracking point on the $\boldsymbol{x_B}$ axis $\varepsilon_x \neq 0$ and $\varepsilon_y = \varepsilon_z = 0$, one would build $\boldsymbol{S}([1/\varepsilon_x\ 0; 0]^\top)$ replacing $1/\varepsilon_y$ and $1/\varepsilon_z$ with 0.

Algorithm 1 Calculation of the Handy matrix $\mathcal{H}$

$\mathcal{H} \leftarrow \mathbb{I}_6$
$\boldsymbol{e} \leftarrow [\varepsilon_x\ \varepsilon_y\ \varepsilon_z]^\top$
$\boldsymbol{\epsilon} \leftarrow [0\ 0\ 0]^\top$
for $k = 1:3$
  if $\boldsymbol{e}(k) \neq 0$
    $\boldsymbol{\epsilon}(k) \leftarrow 1/\boldsymbol{e}(k)$
$\boldsymbol{\Sigma} \leftarrow \boldsymbol{S}(\boldsymbol{\epsilon})$
for $i = 3:6$
  if $\boldsymbol{h_{O_B}}(i) = 1$ **and** $\boldsymbol{h_E}(i) = 0$
    for $j = 1:3$
      if $\boldsymbol{h_E}(j) = 1$ **and** $\boldsymbol{h_{O_B}}(j) = 0$ **and** $j \neq i-3$
        $\mathcal{H}(i,j) \leftarrow \Sigma(i-3,j)$
        $\mathcal{H}(j,:) \leftarrow 0$

Algorithm 1 takes the two vectors $\boldsymbol{h_{O_B}}$ and $\boldsymbol{h_E}$ as inputs as well as the coordinates of the tracking point $\boldsymbol{E}$. The user needs to define the actuated DOF of the vehicle in $\boldsymbol{h_{O_B}}$ and choose, following the design rules evoked before, the controlled DOF at the tracking point in $\boldsymbol{h_E}$.

Overall, algorithm 1 browses through the actuated rotational DOF of the vehicle. If it finds a rotation in $\boldsymbol{h_{O_B}}$ which is not controlled in $\boldsymbol{h_E}$, it checks whether it is used to compensate a translation. When the rotation and translation are found, the appropriate ratio $1/\varepsilon_k$ (ε_k being the k-th coordinate of $\boldsymbol{E}$ in $\mathcal{R}_B$) is selected in $\boldsymbol{\Sigma}$ and placed at the corresponding place in the bottom left submatrix of $\mathcal{H}$. Finally, it cancels the row of $\mathcal{H}$ corresponding to the non actuated translation of $\boldsymbol{h_{O_B}}$.

For demonstration of the matrix generation, let us apply algorithm 1 to the uqr-vessel example (3.18) introduced in the previous section. In this example, the $\mathcal{H}$ matrix must compensate the lack of two translations, sway and heave, with pitch and yaw. Considering a tracking point $\boldsymbol{E}$ of coordinates $[\varepsilon_x\ 0\ 0]^\top$, the skew-symmetrical cross-

product matrix of interest is:

$$\boldsymbol{\Sigma} = \boldsymbol{S}\left(\begin{bmatrix}\frac{1}{\varepsilon_x}\\0\\0\end{bmatrix}\right) = \begin{bmatrix}0 & 0 & 0\\0 & 0 & -\frac{1}{\varepsilon_x}\\0 & \frac{1}{\varepsilon_x} & 0\end{bmatrix}$$

The matrix $\mathcal{H}$ is initialized with the identity.

$$\boldsymbol{h_{O_B}}^1 = \begin{bmatrix}1\\0\\0\\0\\1\\0\end{bmatrix} \quad \boldsymbol{h_E} = \begin{bmatrix}1\\0\\1\\0\\0\\0\end{bmatrix} \quad \mathcal{H} = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\0 & 1 & 0 & 0 & 0 & 0\\0 & 0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 0 & 0\\0 & 0 & 0 & 0 & 1 & 0\\0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

Considering the pair $\boldsymbol{h_{O_B}}$, $\boldsymbol{h_E}$ of (3.18), the first rotational DOF actuated in $\boldsymbol{h_{O_B}}$ but not controlled in $\boldsymbol{h_E}$ is pitch (fifth line, $i = 5$). The algorithm now seeks a translation in $\boldsymbol{h_E}$ which is not actuated in $\boldsymbol{h_{O_B}}$ and can be compensated ($j \neq i - 3$) and finds one on the third line ($j = 3$) corresponding to heave. The appropriate term of $\boldsymbol{\Sigma}$ (second row, third column) is placed in the appropriate position in $\mathcal{H}$ (fifth row, third column) and the heave line of $\mathcal{H}$ is canceled. At this stage $\mathcal{H}$ is:

$$\mathcal{H} = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\0 & 1 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 1 & 0 & 0\\0 & 0 & -\frac{1}{\varepsilon_x} & 0 & 1 & 0\\0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}$$

Then, the algorithm resumes, reading $\boldsymbol{h_{O_B}}$ to find another rotation used for compensation. Yaw is used for compensation of sway ($i = 6$, $j = 4$) so the appropriate term of $\boldsymbol{S}$ is again copied in $\mathcal{H}$ and the sway line of $\mathcal{H}$ is canceled. At this stage the handy matrix for this example is:

$$\mathcal{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{1}{\varepsilon_x} & 0 & 1 & 0 \\ 0 & \dfrac{1}{\varepsilon_x} & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

Finally equation (3.20) shows the handy matrix $\mathcal{H}$ for an uqr-vessel navigating on a (x, y, z) path and with a tracking point at the front of the vehicle. This algorithm can be applied for any suitable pair $\boldsymbol{h_{O_B}}$, $\boldsymbol{h_E}$.

3.3 Application to the *RSM* Robot

In this section, the $\mathcal{H}$-based controller is detailed and adapted to the *RSM* robot. The *RSM* robot is an underactuated underwater vehicle used in several research applications over the years [31, 30]. This vehicle is used in this example as a representative of a class of underactuated marine craft lacking sway and pitch actuation. The propulsive arrangement of this vehicle is described in section 1.2.1.2 and a 3D rendition appears on figure 3.4. The actuation of *RSM* is described here in terms of its associated vectors $\boldsymbol{h_{O_B}}$ and $\boldsymbol{h_E}$. A couple of possible control strategies are described. The complete model-based controller is given in details and a convergence demonstration of the $\mathcal{H}$-based controller is presented.

3.3.1 Actuation and task of the *RSM* Robot

The *RSM* Robot is composed of a cylindrical hull and is equipped with four thrusters. The first two thrusters P_1 and P_2 are in the plane $(\boldsymbol{O_B}, \boldsymbol{x_B}, \boldsymbol{y_B})$ at the rear of the vehicle

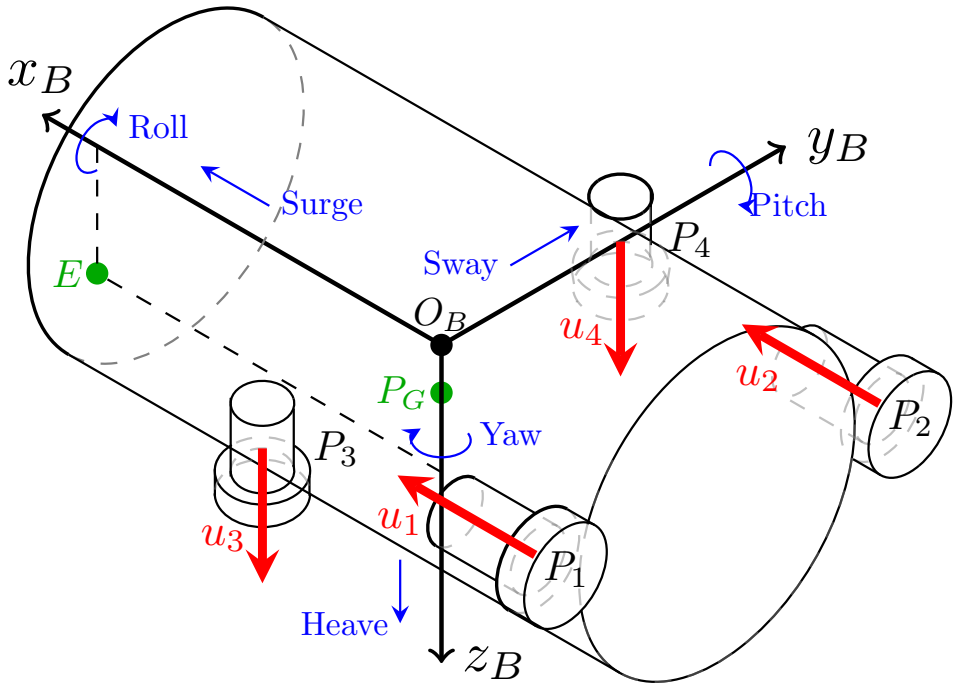


Figure 3.4 – The propulsive configuration of the *RSM* robot

aligned with $\boldsymbol{x_B}$. The two other thrusters are placed in the $(\boldsymbol{O_B}, \boldsymbol{y_B}, \boldsymbol{z_B})$ plane and are aligned with $\boldsymbol{z_B}$. Therefore, the vehicle is able to generate independent surge and heave forces as well as roll and yaw moments. Sway and pitch are not actuated. Nevertheless, because the center of gravity $\boldsymbol{P_G}$ is slightly below the center of buoyancy $\boldsymbol{P_B} = \boldsymbol{O_B}$, the vehicle is naturally stable in roll and pitch.

The actuation vector for *RSM* is:

$$\boldsymbol{h}_{O_B} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \tag{3.21}$$

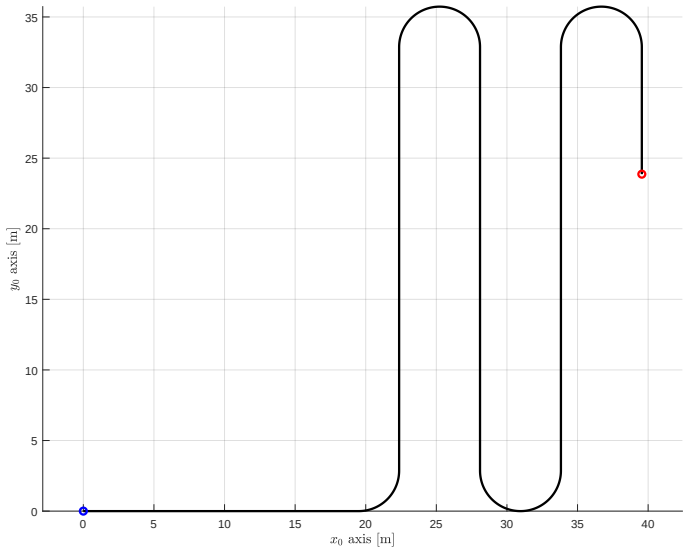


Figure 3.5 – Top view of the seabed scanning trajectory at $1\,\mathrm{m}$ depth.

The tracking point is chosen at the bow of the vehicle below the waterline. The coordinates of $\boldsymbol{E}$ are $[\varepsilon_x \;\; 0 \;\; \varepsilon_z]^\top$ and for this example we take $\varepsilon_x = \varepsilon_z = 0.2\,\mathrm{m}$. As exposed before, the position of the tracking point adds to the stability of the vehicle and is chosen to create exploitable kinematic couplings.

The vehicle is evaluated on the *Seabed Scanning* task presented on figure 3.5. This task is composed of several horizontal rails at constant depth and the vehicle must stay tangent to the rails. Therefore, the required DOF for this task are the three translations and the rotation around $\boldsymbol{z}_0$, the vehicle is ill-actuated. No additional angular constraints are added to the task.

Because the required DOF of the task are different from the actuated DOF of the vehicle, a compensation mechanism is introduced in the controller to cope for the lack of sway actuation and have three actuated translations. Referring to table 3.1, two solutions raise for this tracking point and propulsive arrangement: either use roll and heave or use yaw and surge. This last solution is the most commonly used on usual underactuated surface vessels.

The two strategies are associated with two different vectors for the controlled DOF

at the tracking point:

$$\boldsymbol{h}_{\boldsymbol{E},\boldsymbol{p}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{h}_{\boldsymbol{E},\boldsymbol{r}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

In the first case $\boldsymbol{h}_{\boldsymbol{E},\boldsymbol{p}}$, roll is used for sway compensation so the yaw angle of the vehicle can be controlled. In the second case $\boldsymbol{h}_{\boldsymbol{E},\boldsymbol{r}}$, yaw is used for compensation and can thus not be controlled. In this case though roll, which is not part of the task, could be controlled. Note that, because of the hydrodynamic effects on the hull of the vehicle, the yaw angle is naturally stabilized at non-zero speed and the hull of the vehicle naturally aligns with the velocity vector. Hence, only minimal yaw disturbance appear in this task when it is not controlled. To see a significant difference between the two strategies, additional tests will be conducted with a yaw constraint different from the tangent to the track.

Of course, the two vectors $\boldsymbol{h}_{\boldsymbol{E},\boldsymbol{p}}$ and $\boldsymbol{h}_{\boldsymbol{E},\boldsymbol{r}}$ correspond to two different matrices $\mathcal{H}_p$ and $\mathcal{H}_r$ respectively. Applying algorithm 1 in the two cases with $\boldsymbol{h}_{\boldsymbol{O_B}}$ defined as in (3.21) yields:

$$\mathcal{H}_p = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1/\varepsilon_z & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.22}$$

and

$$\mathcal{H}_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1/\varepsilon_x & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.23}$$

In order to demonstrate the action of these two matrices, they are applied to the relation (3.15) in the same way as in the theoretical example of section 3.1. Inverting equation (3.15) gives:

$$\boldsymbol{\nu} = \boldsymbol{T}^{-1}\boldsymbol{\nu_E} \tag{3.24}$$

The Handy matrices $\mathcal{H}_p$ and $\mathcal{H}_r$ can be applied to this system giving:

$$\boldsymbol{\nu}_1 = \mathcal{H}_p\boldsymbol{T}^{-1}\boldsymbol{\nu_E} \tag{3.25}$$

and

$$\boldsymbol{\nu}_2 = \mathcal{H}_r\boldsymbol{T}^{-1}\boldsymbol{\nu_E} \tag{3.26}$$

The first case is detailed as:

$$\begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1/\varepsilon_z & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & -\varepsilon_z & 0 \\ 0 & 1 & 0 & \varepsilon_z & 0 & -\varepsilon_x \\ 0 & 0 & 1 & 0 & \varepsilon_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_E \\ v_E \\ w_E \\ p_E \\ q_E \\ r_E \end{bmatrix} = \begin{bmatrix} u_E - \varepsilon_z q_E \\ 0 \\ w_E + \varepsilon_x q_E \\ v_E - \varepsilon_x r_E \\ q_E \\ r_E \end{bmatrix} \tag{3.27}$$

Equations (3.27) shows that, because of the off-diagonal term of $\mathcal{H}_p$, the roll speed p is now a function of the sway speed at point $\boldsymbol{E}$, v_E. Same reasoning can be applied

with $\mathcal{H}_r$, making the yaw speed r a function of the sway speed at point $\boldsymbol{E}$, v_E. The next section explains how this new behavior can be used in a controller to allow tracking of a non-actuated DOF.

3.3.2 Complete controller

This section introduces the complete controller used on the *RSM* Robot in association with the $\mathcal{H}$ matrix-based guidance principle. Some variations of this controller have been introduced in chapter 2. The controller is a two-staged cascade system. A velocity reference is calculated with a PI control law and the $\mathcal{H}$ matrix, to cancel the position errors. A model-based proportional effort vector is then proposed to control the speed of the vehicle towards the reference speed.

The control effort vector $\boldsymbol{\tau_c}$ is calculated as:

$$\boldsymbol{\tau_c} = \boldsymbol{M}(\dot{\boldsymbol{\nu}}_c + \boldsymbol{K}(\boldsymbol{\nu}_c - \boldsymbol{\nu})) + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{3.28}$$

where $\boldsymbol{\nu_c}$ is the reference velocity vector calculated with $\mathcal{H}$ and detailed in the following and $\boldsymbol{K}$ is an appropriate proportional gain matrix.

Equation (3.28) shows a typical feedback linearizing structure associated with a proportional controller [65]. The calculation of $\boldsymbol{\tau_c}$ is directly drawn from the dynamic model of the vehicle equation (1.5). Matrices $\boldsymbol{M}$, $\boldsymbol{C}(\boldsymbol{\nu})$ and $\boldsymbol{D}(\boldsymbol{\nu})$ are the mass, Coriolis and centripetal and damping matrices of the model respectively and $\boldsymbol{g}(\boldsymbol{\eta})$ is the vector of gravity and buoyancy effects. The non linear terms of the model are evaluated at the current state of the system ($\boldsymbol{\eta}$, $\boldsymbol{\nu}$) therefore allowing feedback linearization; Applying the control vector $\boldsymbol{\tau_c}$ cancels the non linear terms of the model equation (1.5), leading to the linear closed-loop system:

$$\dot{\boldsymbol{\nu}} = \dot{\boldsymbol{\nu}}_c + \boldsymbol{K}(\boldsymbol{\nu}_c - \boldsymbol{\nu}) \tag{3.29a}$$

$$\boldsymbol{e_\nu} = (\boldsymbol{\nu}_c - \boldsymbol{\nu}) \tag{3.29b}$$

$$\dot{\boldsymbol{e}}_\nu + \boldsymbol{K}\boldsymbol{e_\nu} = 0 \tag{3.29c}$$

The new velocity reference $\boldsymbol{\nu_c}$ is also built as a model-based feedback linearizing controller but, in this case, including the Handy matrix $\mathcal{H}$ to create the compensation behavior. Recalling the kinematic model of the vehicle including the tracking point $\boldsymbol{E}$

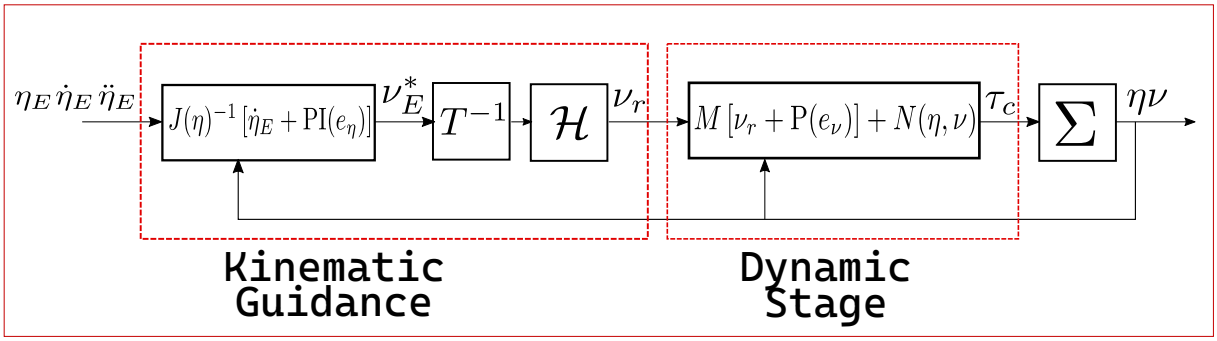


Figure 3.6 – Block diagram of the cascade $\mathcal{H}$-based controller. Σ represents the AUV, $\boldsymbol{N}(\boldsymbol{\eta},\boldsymbol{\nu}) = \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta})$

equation (1.3) one gets:

$$\boldsymbol{\nu_c} = \mathcal{H}\boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\left[\dot{\boldsymbol{\eta}}^* + \boldsymbol{K_p}\boldsymbol{e_\eta} + \boldsymbol{K_i}\int_0^t \boldsymbol{e_\eta}(\zeta)d\zeta\right] \tag{3.30}$$

with $\boldsymbol{e_\eta} = \boldsymbol{\eta}^* - \boldsymbol{\eta_E}$ the error between the desired state $\boldsymbol{\eta}^*$ and the state of the tracking point $\boldsymbol{\eta_E}$.

Overall, this controller is a *cascade system* as depicted on figure 3.6. The inner loop (dynamic stage), calculates the efforts $\boldsymbol{\tau_c}$ required for the vehicle to reach the velocity reference $\boldsymbol{\nu_c}$ calculated by the outer loop (kinematic guidance). The outer loop therefore behaves like a kinematic guidance stage outputting new reference velocities for the controller in a similar way as a LOS guidance stage or any other guidance algorithm would. Cascade systems are known for their natural robustness to external disturbances but the inner loop must be faster than the outer loop to ensure stability.

Note that, in equation (3.30) the PI control law could be replaced by any usual controller (SMC, LQR). Using a different control law in the outer loop could produce different behavior. Notably, adding a sliding surface and switching term would enhance robustness to external disturbances.

3.3.3 Stability proof

To show the convergence of the controller, this section focuses on the second Handy Matrix solution $\mathcal{H}_r$ applied to the *RSM* robot in a planar trajectory tracking application. This example allows demonstrating that the Handy matrix indeed allows convergence of the compensated DOF and does not disturb the other controlled DOF. It could be

reproduced for any other actuation configuration, task, or compensated DOF.

The cascade structure of the controller and the feedback linearizing PI-based forms used in both stages give a first hint towards stability of the controller. Yet, stability of the overall closed-loop system is addressed here to illustrate the behavior of the controller not only with respect to the compensated DOF but also regarding the other DOF not involved in the compensation.

Let us recall the complete model of the underwater vehicle:

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{3.31a}$$

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{T}\boldsymbol{\nu} \tag{3.31b}$$

with, because of underactuation of the *RSM* robot, the vector of propulsive efforts $\boldsymbol{\tau}$ of shape: $\boldsymbol{\tau} = [X\ 0\ Z\ K\ 0\ N]^\top$ (X and Z being non-zero force components and K and M being non-zero moments).

A tracking point $\boldsymbol{E}$ of coordinates $[\varepsilon_x\ 0\ 0]^\top$ in $\mathcal{R}_B$is used in this proof. The associated transformation matrix is:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \varepsilon_x \\ 0 & 0 & 1 & 0 & -\varepsilon_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.32}$$

The sway of the vehicle is not actuated but can be compensated with yaw. The corresponding Handy matrix $\mathcal{H}_r$ calculated with algorithm 1 is given in equation (3.23)

and recalled here:

$$\mathcal{H}_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1/\varepsilon_x & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.33}$$

As detailed in section 3.3.2, the controller used in this example is:

$$\boldsymbol{\tau_c} = \boldsymbol{M}(\dot{\boldsymbol{\nu}}_{\boldsymbol{c}} + \boldsymbol{K}\boldsymbol{e}_{\boldsymbol{\nu}}) + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{3.34a}$$

$$\boldsymbol{\nu_c} = \mathcal{H}_r \boldsymbol{T}^{-1} \boldsymbol{J}(\boldsymbol{\eta})^{-1}(\dot{\boldsymbol{\eta}}^* + \boldsymbol{\lambda}(\boldsymbol{e}_\eta)) \tag{3.34b}$$

In equation (3.34b), $\boldsymbol{\lambda}(\boldsymbol{e}_\eta)$ is a PI control law based on the tracking error $\boldsymbol{e}_\eta$:

$$\boldsymbol{\lambda}(\boldsymbol{e}_\eta) = \boldsymbol{K}_p \boldsymbol{e}_\eta + \boldsymbol{K}_i \int_0^t \boldsymbol{e}_\eta(\zeta) d\zeta \tag{3.35}$$

Following the results of appendix C and as demonstrated before, thanks to the feedback linearizing terms, applying the control vector $\boldsymbol{\tau_c}$ to the dynamic model (3.31a) cancels the non-linearities of the dynamic model on the actuated DOF. The non actuated DOF (sway and pitch) are naturally stable and, because no coupled efforts are generated by the actuators on these two axes, they are assumed to stay stable in closed loop with reasonable disturbances.

Remark 3.1 *Rigorously, the following equations should only be written for the controlled DOF but for the sake of clarity and considering the natural sway and pitch stability of the system, the equations are given on the complete system.*

The effort control vector $\boldsymbol{\tau_c}$ creates the linear closed-loop system:

$$\dot{\boldsymbol{\nu}} = \dot{\boldsymbol{\nu}}_{\boldsymbol{c}} + \boldsymbol{K}(\boldsymbol{\nu_c} - \boldsymbol{\nu}) \tag{3.36}$$

Now, in order to give a demonstration of the convergence of the position errors in $\mathcal{R}_0$, equation (3.36) must be rewritten using the following relations drawn from

equation (3.31b) and equation (3.34b).

$$\dot{\boldsymbol{\nu}} = \boldsymbol{T}^{-1}(\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}}_{\boldsymbol{E}}) \tag{3.37a}$$
$$\dot{\tilde{\boldsymbol{\eta}}} = \dot{\boldsymbol{\eta}}^{*} + \boldsymbol{\lambda}(\boldsymbol{e}_{\eta}) \tag{3.37b}$$
$$\boldsymbol{\nu}_{\boldsymbol{c}} = \mathcal{H}_r\boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\tilde{\boldsymbol{\eta}}} \tag{3.37c}$$
$$\dot{\boldsymbol{\nu}}_{\boldsymbol{c}} = \mathcal{H}_r\boldsymbol{T}^{-1}(\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\tilde{\boldsymbol{\eta}}} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\tilde{\boldsymbol{\eta}}}) \tag{3.37d}$$

The gain matrices of the controller are defined as:

$$\boldsymbol{K} = K\mathbb{I} \tag{3.38a}$$
$$\boldsymbol{K_p} = k_p\mathbb{I} \tag{3.38b}$$
$$\boldsymbol{K_i} = k_i\mathbb{I} \tag{3.38c}$$

where $\mathbb{I}$ is the identity matrix. It implies that the sets of gain parameters are the same on all the DOF. Each independent gain could be tuned to better match the individual characteristics of each DOF without changing the convergence result.

Using the expressions of (3.37) in equation (3.36) gives:

$$\mathcal{H}_r\boldsymbol{T}^{-1}(\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}^{*} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}}^{*}) + \boldsymbol{K}(\mathcal{H}_r\boldsymbol{T}^{-1}\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}^{*} - \boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E}}) = \boldsymbol{T}^{-1}(\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}}_{\boldsymbol{E}}) \tag{3.39}$$

Multiplying both sides by $\boldsymbol{T}$ leads to:

$$\boldsymbol{T}\mathcal{H}_r\boldsymbol{T}^{-1}(\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}^{*} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}}^{*}) + \boldsymbol{T}\boldsymbol{K}(\mathcal{H}_r\boldsymbol{T}^{-1}\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}^{*} - \boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E}}) = \dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}}_{\boldsymbol{E}} \tag{3.40}$$

which can then be rewritten as:

$$\boldsymbol{T}\mathcal{H}_r\boldsymbol{T}^{-1}(\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}^{*} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}}^{*}) + \boldsymbol{K}(\boldsymbol{T}\mathcal{H}_r\boldsymbol{T}^{-1}\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}^{*} - \boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E}}) = \dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} + \boldsymbol{J}(\boldsymbol{\eta})^{-1}\ddot{\boldsymbol{\eta}}_{\boldsymbol{E}} \tag{3.41}$$

The matrix product $\boldsymbol{T}\mathcal{H}_r\boldsymbol{T}^{-1}$ reads:

$$\boldsymbol{T}\mathcal{H}_r\boldsymbol{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1/\varepsilon_x & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.42}$$

and, considering that the roll and pitch are stable and undisturbed, the vehicle is supposed to stay in the horizontal plane where $\phi \approx 0$ and $\theta \approx 0$. Matrices $\boldsymbol{J}(\boldsymbol{\eta})^{-1}$ and $\dot{\boldsymbol{J}}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})^{-1}$ are then:

$$\boldsymbol{J}(\boldsymbol{\eta})^{-1} = \begin{bmatrix} c\psi & s\psi & 0 & 0 & 0 & 0 \\ -s\psi & c\psi & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.43a}$$

$$\dot{\boldsymbol{J}}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})^{-1} = \begin{bmatrix} -\dot{\psi}s\psi & \dot{\psi}c\psi & 0 & 0 & 0 & 0 \\ -\dot{\psi}c\psi & -\dot{\psi}s\psi & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.43b}$$

Considering equation (3.42) and equation (3.43) and after some calculations, the two first lines of equation (3.41) give in the earth-fixed frame:

$$\ddot{e}_x + (k_p + K)\dot{e}_x + (k_i + Kk_p)e_x + Kk_i\chi_x + \dot{\psi}(\dot{e}_y + k_p e_y + k_i\chi_y) = 0 \tag{3.44a}$$
$$\ddot{e}_y + (k_p + K)\dot{e}_y + (k_i + Kk_p)e_y + Kk_i\chi_y - \dot{\psi}(\dot{e}_x + k_p e_x + k_i\chi_x) = 0 \tag{3.44b}$$

with $\chi_x = \int_0^t e_x(\zeta)d\zeta$ and $\chi_y = \int_0^t e_y(\zeta)d\zeta$.

Remark that the sixth equation of (3.41) gives the dynamics of the orientation parameter ψ, not controlled in this case:

$$\ddot{\psi} = \frac{1}{\varepsilon_x}\left[-\dot{\psi}(\dot{x}^* c\psi + \dot{y}^* s\psi) - \ddot{x}s\psi + \ddot{y}c\psi + K(-\dot{x}^* s\psi + \dot{y}^* c\psi)\right] - K\dot{\psi} \tag{3.45}$$

Equation (3.45) has the shape of a Liénard equation whom study is outside of the scope of this work.

3.3.3.1 Lyapunov argument

The stability of the closed-loop system (3.44) is demonstrated in this Section using Lyapunov theory. First, new error variables are introduced:

$$\sigma_x = \dot{e}_x + k_p e_x + k_i\chi_x \tag{3.46a}$$
$$\sigma_y = \dot{e}_y + k_p e_y + k_i\chi_y \tag{3.46b}$$

The two error variables σ_x and σ_y are similar to the standard definition of sliding surfaces (see section 2.4) but are just used as intermediate variables in the present demonstration. The first order derivatives of these two new error variables are:

$$\begin{aligned}\dot{\sigma}_x &= -(k_p + K)\dot{e}_x - (k_i + Kk_p)e_x - Kk_i\chi_x - \dot{\psi}(\dot{e}_y + k_p e_y + k_i\chi_y) + k_p\dot{e}_x + k_i e_x \\ &= -K\sigma_x - \dot{\psi}\sigma_y \end{aligned} \tag{3.47a}$$
$$\begin{aligned}\dot{\sigma}_y &= -(k_p + K)\dot{e}_y - (k_i + Kk_p)e_y - Kk_i\chi_y + \dot{\psi}(\dot{e}_x + k_p e_x + k_i\chi_x) + k_p\dot{e}_y + k_i e_y \\ &= -K\sigma_y + \dot{\psi}\sigma_x \end{aligned} \tag{3.47b}$$

Remark 3.2 *At this point and to stick with the concept of SMC, choosing an appropriate sliding condition would also allow demonstrating that an additional SMC-based stage of control would add robustness to external bounded disturbances.*

A natural Lyapunov function candidate is:

$$V = \frac{1}{2}(\sigma_x^2 + \sigma_y^2) \tag{3.48}$$

Function V is of class $\mathcal{C}^1$, positive definite and radially unbounded. The first order time derivative of the Lyapunov function candidate is:

$$\dot{V} = \dot{\sigma}_x\sigma_x + \dot{\sigma}_y\sigma_y = -K\sigma_x^2 - \dot{\psi}\sigma_x\sigma_y - K\sigma_y^2 + \dot{\psi}\sigma_x\sigma_y = -K(\sigma_x^2 + \sigma_y^2) \tag{3.49}$$

Then, for any choice of K strictly positive, $\dot{V}$ is strictly negative and $\dot{V} \leq -\alpha V$ with $\alpha \geq 2K$. Consequently, the equilibrium point $\sigma_x = 0$, $\sigma_y = 0$ is globally exponentially stable (GES). Also, owing to the cascade structure of equation (3.46), the equilibrium $e_x = 0$, $e_y = 0$ is then GES too.

One interesting remark is that the GES character of the method is independent from any condition on the yaw rate of the vehicle. This property is of particular interest in controlling the AUV planar trajectory, considering that the dynamics of ψ given by equation (3.45) are hardly predictable or controllable.

3.4 Simulation results

This section shows trajectory tracking results obtained in simulation on the *RSM* robot using the controller based on the Handy matrix $\mathcal{H}$. Several tests are conducted; First, the two possible compensation solutions $\mathcal{H}_p$ and $\mathcal{H}_r$ (section 3.3.1) are compared on the Seabed Scanning task. Then, the $\mathcal{H}_r$ solution is compared with a traditional LOS controller and a SM controller on the same task with and without sea current.

3.4.1 Comparison of compensation solutions

3.4.1.1 Seabed Scanning

First, the two different compensation solutions $\mathcal{H}_p$ and $\mathcal{H}_r$ are compared on the seabed scanning task using a PI control law in the outer loop and a P control law in the inner loop. table 3.2 gives the simulation parameters used for this experiment as well as the position of the tracking point and the gain parameters used for both vehicles.

Item	Value	Item	Value
Trajectory	Seabed Scanning	**Nominal Speed**	$1.5m.s^{-1}$
Current velocity	$[0\ 0\ 0]^\top$	**Simulation time sample**	$0.05ms$
Vehicle Shape	Cylinder	**Propulsive configuration**	*RSM*
Tracking point	$[0.2\ 0\ 0.2]^\top$	K	8
k_p	8	k_i	2

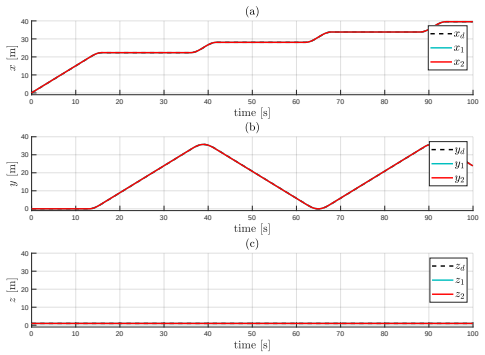
Table 3.2 – Simulation parameters - seabed scanning mission

Figure 3.7 displays the position and orientation of both vehicles with each compensation strategy and shows that both solutions have equally good position tracking performances on the three axes. This confirms that the method produces the desired results in terms of position tracking regardless of the chosen compensation strategy. However, figure 3.7 shows differences in the orientation of both vehicles as highlighted in figure 3.8.

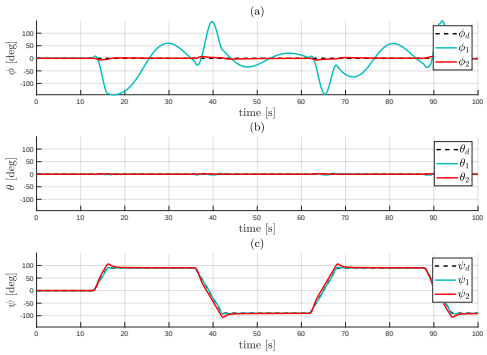
This last figure, compares the orientation errors between the two solutions. Here, notable differences raise. First, figure 3.8.(a) shows that the first compensation strategy creates a larger roll disturbance than the second one. This results was expected since this DOF is used for compensation in the first case and can therefore not be controlled while it is controlled in the second case. Yet, this greater disturbance is acceptable since roll is not part of the required DOF of the task. For the same reason, figure 3.8.(c) shows a larger error on the yaw angle when the second solution is used. In this case, yaw is used for compensation and is not controlled. It leads to approximately $25°$ errors in curves. This error is naturally canceled on the straight lines because of the hydrodynamic drag which realigns the vehicle on the global speed.

3.4.1.2 Rail on the y_o axis with yaw constraint

The differences between the two strategies appear more clearly on figure 3.9. This time the vehicle is evaluated on a single rail aligned with the $\boldsymbol{y_o}$ axis at constant depth but with the additional yaw constraint of staying perpendicular to the trajectory. The desired yaw angle is $\psi^* = 0$ during all the task.Once again, the method shows very good tracking results in position for both strategies. However, the first vehicle using matrix



(a) Position comparison.
a: $\boldsymbol{x_o}$ axis, b: $\boldsymbol{y_o}$ axis, c: $\boldsymbol{z_o}$ axis



(b) Orientation comparison.
a: Roll, b: Pitch, c: Yaw

Figure 3.7 – Comparison of the position and orientation of the two vehicles. Blue: $\mathcal{H}_p$, Red: $\mathcal{H}_r$, Dashed: Reference

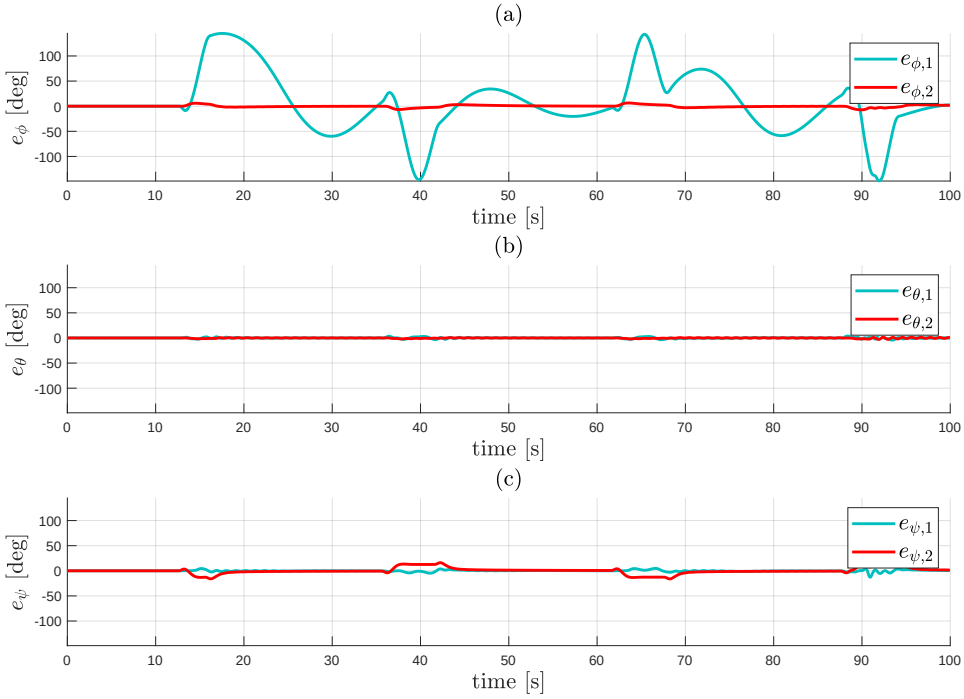


Figure 3.8 – Comparison of orientation errors for the two compensation solutions on the Seabed Scanning. Blue: $\mathcal{H}_p$, Red: $\mathcal{H}_r$ - a: Roll error - b: Pitch error - c: Yaw error

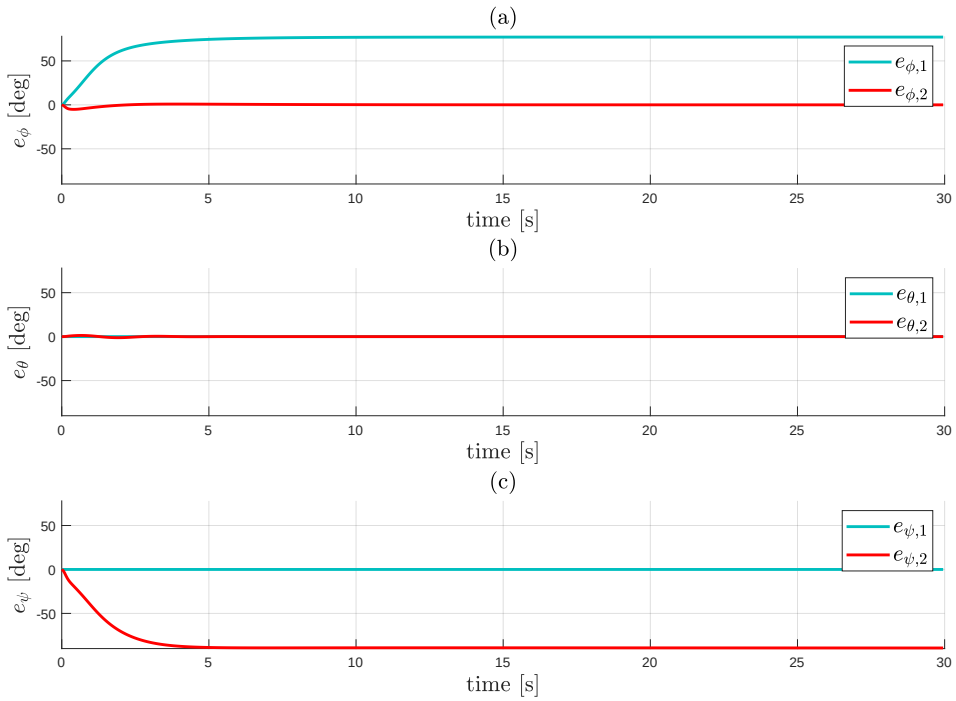


Figure 3.9 – Comparison of orientation errors for the two compensation solutions on the $\boldsymbol{y_o}$ rail with $\psi^* = 0$. Blue: $\mathcal{H}_p$, Red: $\mathcal{H}_r$ - a: Roll error - b: Pitch error - c: Yaw error

$\mathcal{H}_p$ generates a roll angle of approximately $70°$ degrees but has a perfect yaw tracking while the second vehicle using matrix $\mathcal{H}_r$ generates a yaw error of $90°$. Here again the second vehicle neglects the desired yaw and stabilizes along the linear velocity vector of the trajectory.

This last case study shows two different things. First, it demonstrates that some strategies can provide completely different results on different tasks. In this case, the yaw compensation solution using $\mathcal{H}_r$ can be considered acceptable on the first task (figure 3.7) but is totally not suited for the second task (figure 3.9). It demonstrates that the combination of trajectory, constraints and natural dynamics of the vehicle is central in the definition of the task. In the first task, the yaw angle is only slightly disturbed by the controller and hydrodynamic effects cause it to tend naturally towards the desired value. In the definition of the trajectory, choosing a desired yaw angle equal to the tangent to the desired velocity vector does match the natural dynamics of the vehicle. Depending on the application, a little disruption of the yaw angle during the turns can be accepted. On the second task however, the yaw constraint goes radically against the natural dynamics of the vehicle. In this case, the yaw angle of the vehicle must be actively controlled to meet the constraints imposed by the task. Comparing these two task shows that, even if they theoretically require the same DOF (surge, sway, heave and yaw), the association of the trajectory constraints and natural dynamics of the chosen vehicle implies two different control strategies.

The second effect demonstrated by these two case studies is the possible disturbance created by the chosen compensation strategy. The yaw angle constraint has been added to keep the vehicle pointed perpendicularly to a vertical plane of a virtual object in the spirit of a sensor inspection of a submerged structure. In such a case, the constraint is not really on the yaw angle of the vehicle but is rather on the orientation of the $\boldsymbol{x_B}$ axis of the vehicle with respect to an object in the inertial frame. Thus, to maintain the desired orientation, the vehicle must be able to control the orientation of its longitudinal axis around the vertical axis of the inertial plane, $\boldsymbol{z_0}$. Yet, figure 3.9 shows that the roll compensation strategy can create quite drastic roll angles depending on the required sway and heave speeds. Here, the roll of the vehicle stabilizes around $70°$ but the value of the equilibrium depends on the desired speeds of the trajectory. For a greater desired speed, if the vehicle had a slightly lower buoyancy or a center of gravity closer to its center of buoyancy, the roll equilibrium found by the controller would tend to $90°$. The issue is that, at $\phi = 90°$, the *RSM* propulsive arrangement is unable to generate

any moment around the vertical axis of the inertial frame z_o because the two rear thrusters end up in a vertical plane. Despite the fact that the vehicle and control strategy theoretically allow good control of all the required DOF of the task, the choice of the compensation strategy can create attitudes in which the vehicle looses control over one or several other DOF. This case of study may seem far fetched but similar issues can appear with other control laws and DOF.

3.4.2 Comparison with LOS and SMC

In this section, the controller based on the $\mathcal{H}_r$ matrix is compared with a Sliding Mode Controller (SMC) drawn from [40] and a PID controller associated with Line Of Sight guidance (LOS) derived from [105]. The three controllers are evaluated on the Seabed Scanning task first without external disturbance and then in presence of an unknown constant irrotational sea current. The three controllers are applied to the *RSM* vehicle.

The LOS and SMC controllers are some of the most common in literature. They are presented in detail in section 2.1 and section 2.4 respectively. As with the $\mathcal{H}_r$ matrix, both controllers can be used to compensate the lack of sway with yaw.

The structures of the controllers are slightly different. The LOS and $\mathcal{H}_r$ based controllers are cascade system composed of a guidance stage as the outer loop and a feedback linearizing proportional controller as the inner loop. Whereas the SMC controller is built in a single stage. The yaw sliding surface used in the calculation of the yaw input is a function of the sway error. Equations of the dynamic model are used to calculate the input through dynamic couplings. None of the three controller control the yaw angle of the vehicle.

Table 3.3 shows the simulation parameters used in the two experiments. The gain parameters of the three controllers are the same in the two tests to evaluate robustness against unpredictable disturbances.

3.4.2.1 Without external disturbance

Figure 3.10 shows that the position tracking performance of the three controllers are virtually identical without sea current. Figure 3.11 shows that the compensation strategies used in the three controller do not destabilize the naturally stable DOF of the three vehicles. Notably, the yaw angle of the three vehicles naturally tends towards the

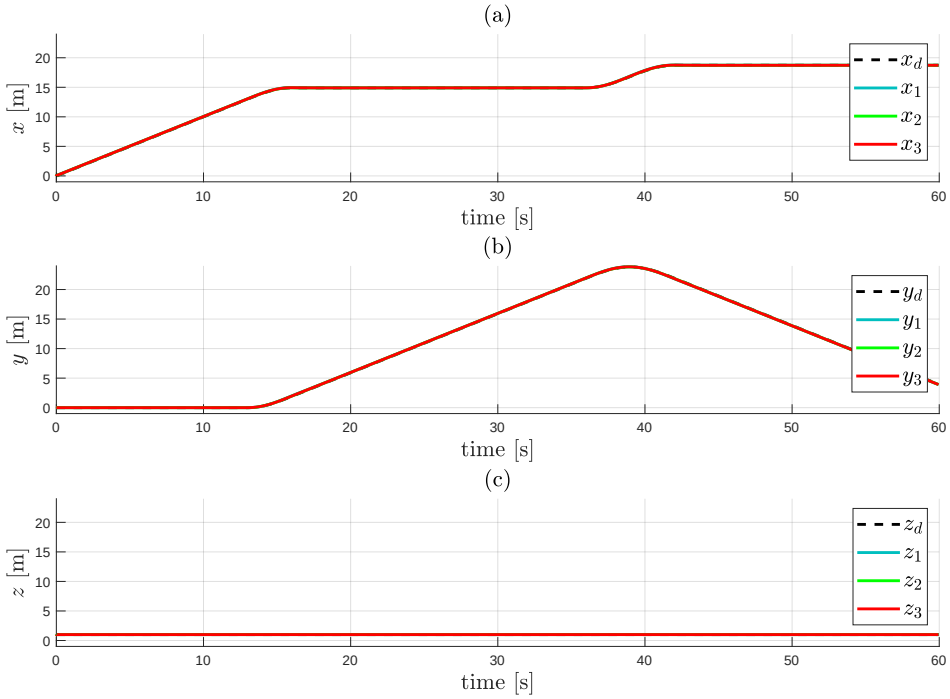


Figure 3.10 – Comparison of position in $\mathcal{R}_0$for the LOS controller (Blue), the SMC (Green) and the $\mathcal{H}$ controller (Red) on the Seabed Scanning task without current. a: $\boldsymbol{x_o}$ axis - b: $\boldsymbol{y_o}$ axis - c: $\boldsymbol{z_o}$ axis

tangent to the track in between the turns and is only disturbed to correct sway errors in the turns.

3.4.2.2 Constant sea current

In the presence of a constant sea current, the tracking performance is degraded for both controllers as shown on Figs. 3.12 and 3.13. However, the performance degradation of the first vehicle (mH with kinematic PI) is probably related more to the use of a PI control law than to the guidance method. The response of the PI being quite slow, the vehicle falls behind during the turns (around 16s, 44s, 68s and 92s) but quickly catches back with the desired trajectory. All the controlled DOF converge on the trajectory between the turns showing that the guidance principle stays effective in the presence of sea current.

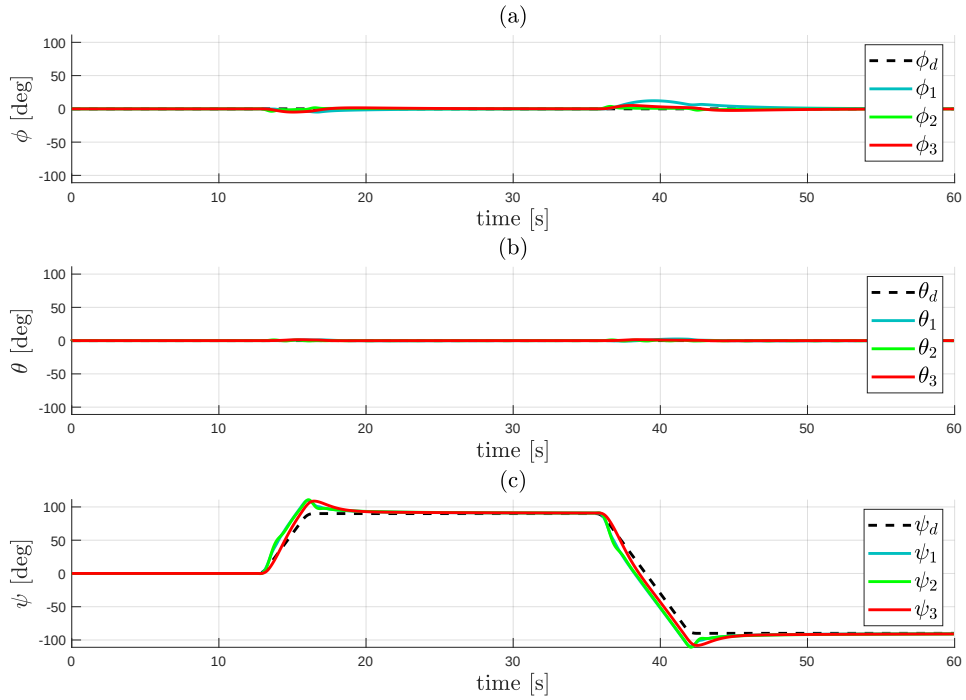


Figure 3.11 – Comparison of orientation for the LOS controller (Blue), the SMC (Green) and the $\mathcal{H}$ controller (Red) on the Seabed Scanning task without current. a: Roll - b: Pitch - c: Yaw

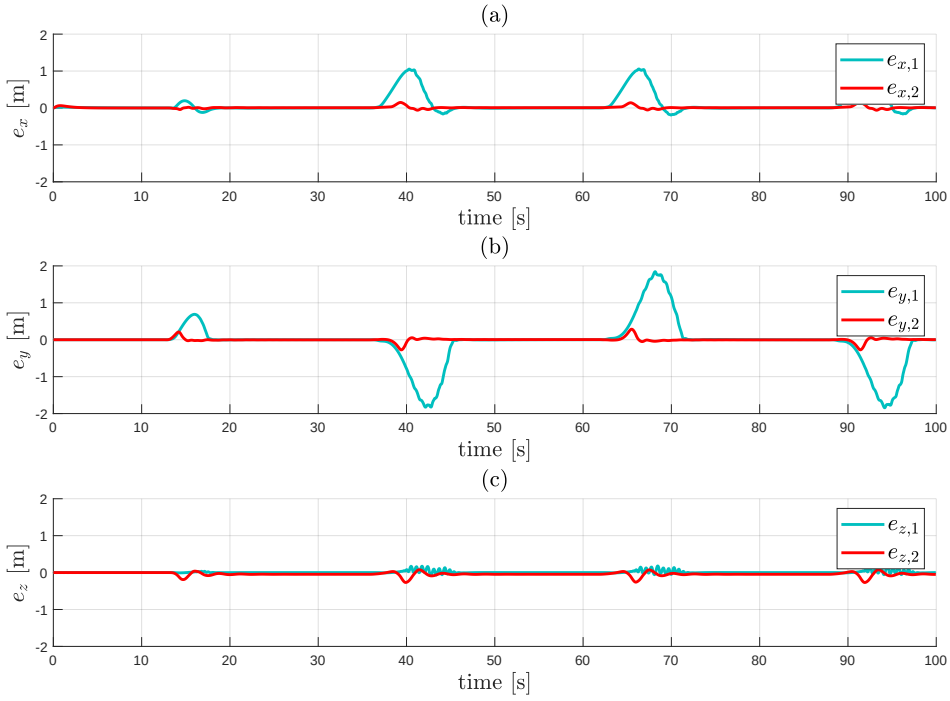


Figure 3.12 – Comparison of position errors for the $\mathcal{H}$ controller (1, Blue) and the SM controller (2, Red) on the Seabed Scanning with constant current on $\boldsymbol{x_o}$ axis. a: Error on $\boldsymbol{x_o}$ - b: Error on $\boldsymbol{y_o}$ - c: Error on $\boldsymbol{z_o}$

The SM controller of the second vehicle on the other hand (Red line on Figs. 3.12 and 3.13) has a more aggressive response during the turns which limits the error but can experience steady state error as on figure 3.13.(a).

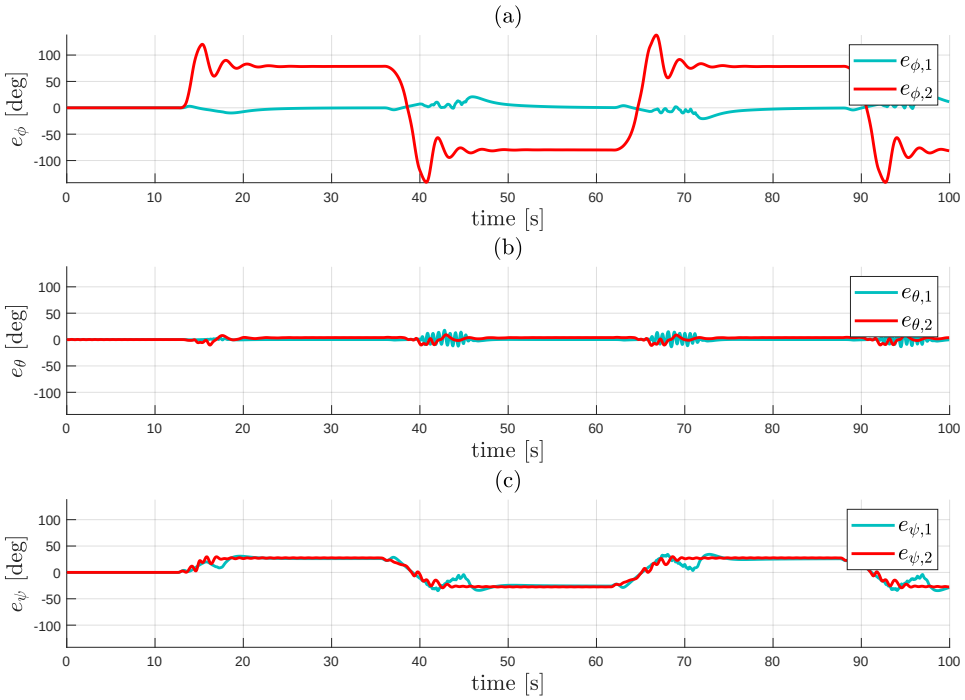


Figure 3.13 – Comparison of orientation errors for the $\mathcal{H}$ controller (1, Blue) and the SM controller (2, Red) on the Seabed Scanning with constant current on $\boldsymbol{x_o}$ axis. a: Roll error - b: Pitch error - c: Yaw error

Item	**Value**	**Item**	**Value**
Trajectory	Seabed Scanning	**Nominal Speed**	$1.5m.s^{-1}$
Current velocity	$[-0.75m.s^{-1}\ 0\ 0]^\top$	**Simulation time sample**	$0.05ms$
Vehicle Shape	Cylinder	**Propulsive arrangement**	*RSM*
$\mathcal{H}_r$ based controller			
Tracking point	$[0.2\ 0\ 0]^\top$	K	8
k_p	8	k_i	2
LOS Guidance			
k_u	60	$k_{i,u}$	3
$k_{d,u}$	1	k_r	20
$k_{i,r}$	1	$k_{d,r}$	10
K_p	20	K_i	5
K_d	8		
Sliding Mode Controller			
l_x	1	l_y	1
k_x	1	k_y	1
λ_1	5	λ_2	5
λ_3	1	K_u	2
K_r	2	W_u	1
W_r	1		

Table 3.3 – Simulation parameters, with sea current

3.5 Path-Following application of the $\mathcal{H}$ matrix controller

In this section, the $\mathcal{H}$ matrix controller is applied to a path-following problem. The Remus100 AUV has been used for this illustration and the task is a mock-up of a typical mission. It is made up of a series of waypoints joined by straight routes.

As described briefly in section 1.4.2.2, in the path-following scenario, the vehicle is constrained to the path between the waypoints. A virtual particle is simulated on the path, evolving according to the behavior of the vehicle so that it can never leave the actual particle representing the vehicle behind. The path-following approach used in this example is adapted from [20] and similar work can be found for instance in [111].

In path following, the geometric and dynamic tasks are tackled independently. In this example however, the solution given by the $\mathcal{H}$ controller is demonstrated to solve the dynamic task and provide a general solution to the geometric task at the same time.

3.5.1 The Remus100 AUV

The Remus100 (Remote Environmental Monitoring UnitS) shown on figure 3.14 is a commercial AUV designed by the Oceanographic System Lab. of the Woods Hole Oceanographic Institution and used in applications of hydrographic surveys, scientific sampling or pipeline inspections. It has been used in many research and industry applications and has a very well established model.

The Remus100 is equipped with a fixed rear thrusters, a rudder and a stern plane. The actuation configuration of the vehicle generates an independent surge force and pitch and yaw moments depending on the relative velocity of the vehicle and the surrounding water. Following the remark made on coupled DOF in section 5.1, the sway and heave forces generated by the stern control surfaces are neglected in the controller calculations. A simplified allocation model has been used to calculate the control sur-

Figure 3.14 – Remus100 (Image from *OSL WHOI*)

faces defection angles and the necessary propeller speed.

3.5.2 $\mathcal{H}$ matrix controller for Remus100

The $\mathcal{H}$ matrix controller for the Remus100 vehicle is introduced in this section. The task is focused only on the position of the vehicle, the attitude of the vehicle is stabilized by hydrodynamic effects and hydrostatic restoring moments acting on the hull and control surfaces. As evoked before, the vehicle is actuated in surge, pitch and yaw.

Recalling the DOF representation introduced in section 3.2.2, the actuated DOF of the vehicle at the center of the mobile frame are represented as:

$$\boldsymbol{h}_{\boldsymbol{O_B}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}^\top \tag{3.50}$$

To allow compensation of the lack of actuation in sway and heave, the tracking point is placed at the nose of the vehicle, $E = [\varepsilon_x, 0, 0]^\top$ with ε_x approximately equal to half the hull length of the AUV. Having the tracking point at the front of the vehicle also increases the natural pitch and yaw stability. The controlled DOF at the tracking point are represented as:

$$\boldsymbol{h}_{\boldsymbol{E}} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^\top \tag{3.51}$$

The Remus100 vehicle associated with the position task of this example mtch with the example given in section 3.2.3. Using algorithm 1 with $\boldsymbol{h}_{\boldsymbol{O_B}}$ and $\boldsymbol{h}_{\boldsymbol{E}}$ defined in equation (3.50) and (3.51) respectively, the $\mathcal{H}$ matrix is calculated as:

$$\mathcal{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{1}{\varepsilon_x} & 0 & 1 & 0 \\ 0 & \dfrac{1}{\varepsilon_x} & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.52}$$

Then, using the equations introduced in chapter 3, the $\mathcal{H}$ matrix controller is built as a

cascaded model-based linearizing controller:

$$\boldsymbol{\nu_c} = \mathcal{H}\boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\boldsymbol{\lambda}(\boldsymbol{\eta_E}, \dot{\boldsymbol{\eta}}_{\boldsymbol{E}}, \alpha) \tag{3.53a}$$

$$\boldsymbol{\tau_c} = \boldsymbol{M}(\dot{\boldsymbol{\nu}}_{\boldsymbol{c}} + \boldsymbol{K}\boldsymbol{e}_{\boldsymbol{\nu}}) + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{3.53b}$$

where α is an auxiliary variable to be defined later and $\boldsymbol{\lambda}(\boldsymbol{\eta_E}, \dot{\boldsymbol{\eta}}_{\boldsymbol{E}}, \alpha)$ is a control law to be determined. Because of the path following scenario, the control law will be a function of the current state of the vehicle and of the current state of the virtual path particle contained in α which is not known a priori but will be used as a control parameter and defined in the following. The non linear terms of the model are canceled using feedback linearization. The matrix $\boldsymbol{K}$ is a conventional positive definite proportional gain matrix and $\boldsymbol{e}_{\boldsymbol{\nu}} = \boldsymbol{\nu_c} - \boldsymbol{\nu}$.

A complete proof of the stability of the $\mathcal{H}$-based controller can be found in chapter 3. Nevertheless, the cascaded structure of the system allows a rapid convergence demonstration. The feedback linearizing structure of the dynamic part of the controller equation (3.53b) gives the closed-loop system:

$$\dot{\boldsymbol{\nu}} = \dot{\boldsymbol{\nu}}_{\boldsymbol{c}} + \boldsymbol{K}\boldsymbol{e}_{\boldsymbol{\nu}} \quad \Rightarrow \quad \boldsymbol{\nu} \to \boldsymbol{\nu_c} \tag{3.54}$$

Remark 3.3 *Here, as in section 3.3.3, the stability demonstration is given on the full six DOF state space. The stability of the non actuated DOF is assumed considering the great number of applications of the Remus100 vehicle where possible instability of the non actuated DOF due to coupling effects are not taken into account. A more rigorous demonstration should demonstrate the stability of the actuated DOF first, then show that the disturbance generated on the non actuated DOF does not make them unstable.*

Then, providing that the inner loop is faster than the outer loop, applying equation (3.53a) to equation (1.3) gives the closed-loop system:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{T}\mathcal{H}\boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\boldsymbol{\lambda}(\boldsymbol{\eta_E}, \dot{\boldsymbol{\eta}}_{\boldsymbol{E}}, \alpha) \tag{3.55}$$

with:

$$\boldsymbol{T}\mathcal{H}\boldsymbol{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\dfrac{1}{\varepsilon_x} & 0 & 0 & 0 \\ 0 & \dfrac{1}{\varepsilon_x} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.56}$$

Breaking down the velocity vector into translation and rotation components $\dot{\boldsymbol{\eta}}_{\boldsymbol{E}} = [\dot{\boldsymbol{\eta}}_{\boldsymbol{E}1}^\top, \dot{\boldsymbol{\eta}}_{\boldsymbol{E}2}^\top]^\top$ and the control law in the same way, $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1^\top, \boldsymbol{\lambda}_2^\top]^\top$, the closed-loop system (3.55) gives:

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{E}1} \equiv \boldsymbol{\lambda}_1 \tag{3.57}$$

The $\boldsymbol{\lambda}(\boldsymbol{\eta}_{\boldsymbol{E}}, \dot{\boldsymbol{\eta}}_{\boldsymbol{E}}, \alpha)$ function must therefore be chosen to solve the path following problem described in the following.

3.5.3 The path-following problem

In this section, the path-following problem is described using the formalism of [20]. The position of the particle representing the tracking point of the vehicle is denoted $\boldsymbol{p}$ and the position of the virtual particle evolving on the path is denoted $\boldsymbol{p}_{\boldsymbol{p}}$. The geometric path is parametrized by the scalar variable ϖ. It is defined as:

$$P = \{\boldsymbol{p} \in \mathbb{R}^3 \ \mid \ \boldsymbol{p} = \boldsymbol{p}_{\boldsymbol{p}}(\varpi) \ \ \forall\, \varpi \in \mathbb{R}\} \tag{3.58}$$

The geometric path is defined under the following assumptions motivated by the stability results described later in this section (see equation (3.70)):

A.1 The path is regular: $\varpi > 0\ ,\ \dot{\varpi} > 0\ \forall\, t > 0$.

A.2 The virtual particle is only moving forward on the path with the velocity vector $\boldsymbol{v}_{\boldsymbol{p}} = [U_p, 0, 0]^\top$ expressed in the path frame with $U_p > 0$.

The parameter ϖ can be seen as the curvilinear abscissa of the virtual particle on the path. The propagation of ϖ will indirectly be used as a control parameter in the following to create the desired following behavior and move the virtual particle according

to the evolution of the vehicle. A local path frame $\mathcal{R}_p$centered on $\boldsymbol{p_p}(\varpi)$ is defined such that the $\boldsymbol{x_p}$ axis of $\mathcal{R}_p$is tangent to the path at all times. The path frame is the result of two consecutive rotations of the inertial frame. First a rotation of angle χ_p around the $\boldsymbol{z_o}$ axis and then a rotation of angle ν_p around the $\boldsymbol{y}$ axis of the intermediate frame. The two angles are defined for a given $\boldsymbol{p_p}(\varpi)$ as:

$$\chi_p(\varpi) = \arctan\left(\frac{y'_p(\varpi)}{x'_p(\varpi)}\right) \tag{3.59a}$$

$$\nu_p(\varpi) = \arctan\left(\frac{-z'_p(\varpi)}{\sqrt{x'_p(\varpi)^2 + y'_p(\varpi)^2}}\right) \tag{3.59b}$$

where $\cdot'$ is the derivative operator $\dfrac{d}{d\varpi}$. The two associated rotation matrices are given as:

$$\boldsymbol{R_{p,z}}(\chi_p(\varpi)) = \begin{bmatrix} \cos(\chi_p) & -\sin(\chi_p) & 0 \\ \sin(\chi_p) & \cos(\chi_p) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.60a}$$

$$\boldsymbol{R_{p,y}}(\nu_p(\varpi)) = \begin{bmatrix} \cos(\nu_p) & 0 & \sin(\nu_p) \\ 0 & 1 & 0 \\ -\sin(\nu_p) & 0 & \cos(\nu_p) \end{bmatrix} \tag{3.60b}$$

The full rotation from the inertial frame to the path frame is hence defined as:

$$\boldsymbol{R_p}(\varpi) = \boldsymbol{R_{p,z}}(\chi_p(\varpi))\boldsymbol{R_{p,y}}(\nu_p(\varpi)) \tag{3.61}$$

The following error between the virtual particle on the path and the tracking point of the vehicle is then defined in the path frame:

$$\boldsymbol{\varepsilon} = \boldsymbol{R_p}^\top(\boldsymbol{p} - \boldsymbol{p_p}(\varpi)) \tag{3.62}$$

Note that, in order to stick to the formalism of [20], the error convention used in this section is the opposite of the convention used elsewhere in this work. The error vector

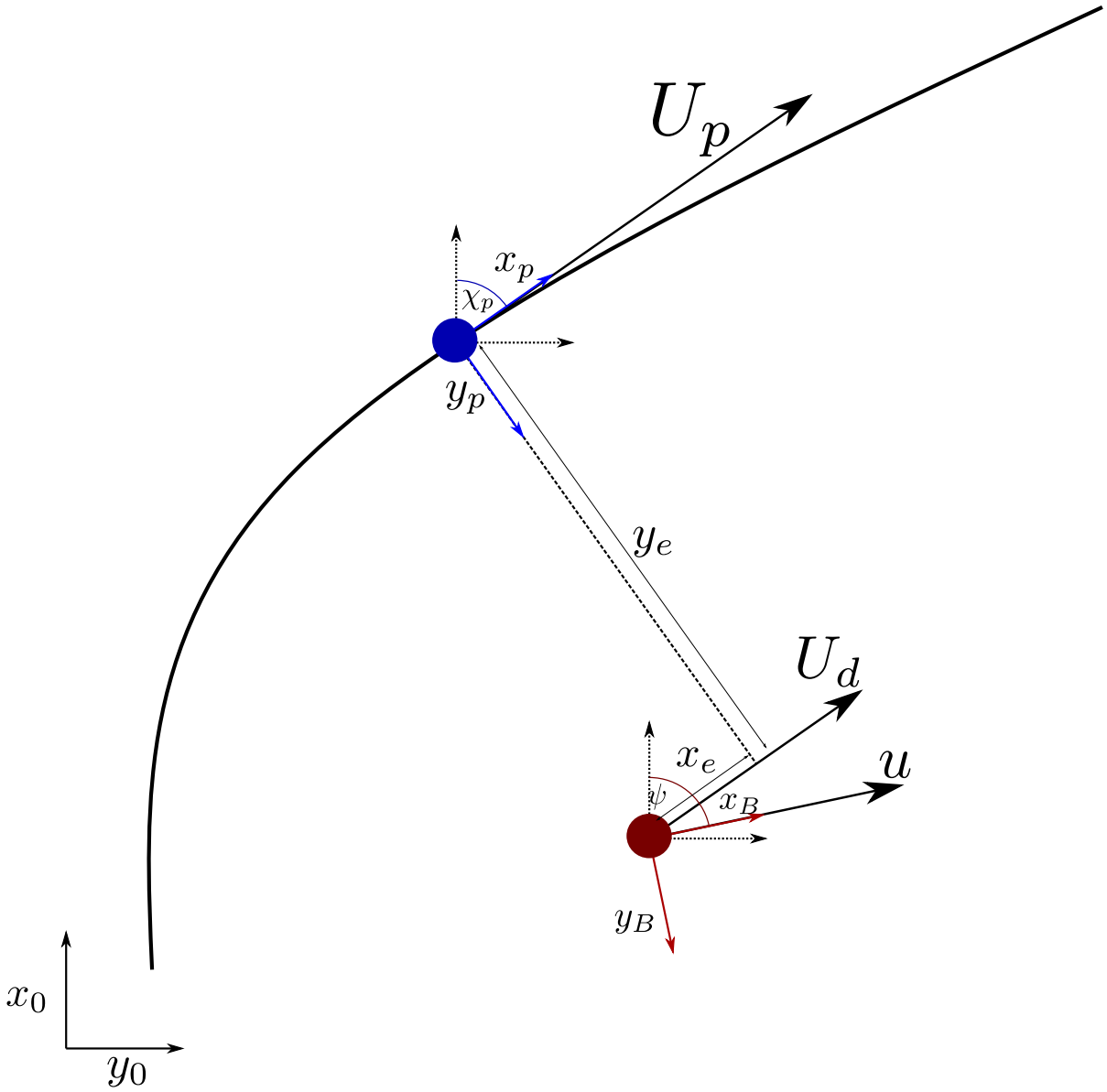


Figure 3.15 – Path following problem in the horizontal plane. Blue : Path particle, Red : Vehicle

$\boldsymbol{\varepsilon} = [x_e, y_e, z_e]^\top$ is broken down into the *along-track error* x_e, *cross-track error* y_e and *vertical-track error* z_e. The path-following problem thus becomes the error minimizing problem of driving the error vector $\boldsymbol{\varepsilon}$ to zero.

The path following problem is once again solved using Lyapunov theory. Consider the Lyapunov function candidate V_ε:

$$V_\varepsilon = \frac{1}{2}\boldsymbol{\varepsilon}^\top\boldsymbol{\varepsilon} \tag{3.63}$$

The first-order time derivative of the error vector expressed in the path frame is:

$$\begin{aligned}\dot{\boldsymbol{\varepsilon}} &= \dot{\boldsymbol{R}_p}^\top(\boldsymbol{p} - \boldsymbol{p_p}) + \boldsymbol{R_p}^\top(\dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_{\boldsymbol{p}}) \\ &= \boldsymbol{S_p}^\top\boldsymbol{R_p}^\top(\boldsymbol{p} - \boldsymbol{p_p}) + \boldsymbol{R_p}^\top(\dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_{\boldsymbol{p}}) \\ &= \boldsymbol{S_p}^\top\boldsymbol{\varepsilon} + \boldsymbol{R_p}^\top(\dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_{\boldsymbol{p}})\end{aligned} \tag{3.64}$$

with the skew-symmetric matrix $\boldsymbol{S_p} = -\boldsymbol{S_p}^\top$ is:

$$\boldsymbol{S_p}^\top = \begin{bmatrix} 0 & -\dot{\chi}_p\cos(\nu_p) & \dot{\nu}_p \\ \dot{\chi}_p\cos(\nu_p) & 0 & \dot{\chi}_p\sin(\nu_p) \\ -\dot{\nu}_p & -\dot{\chi}_p\sin(\nu_p) & 0 \end{bmatrix} \tag{3.65}$$

The first-order time derivative of the Lyapunov candidate V_ε becomes:

$$\begin{aligned}\dot{V}_\varepsilon &= \boldsymbol{\varepsilon}^\top\left(\boldsymbol{S_p}^\top\boldsymbol{\varepsilon} + \boldsymbol{R_p}^\top(\dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_{\boldsymbol{p}})\right) \\ &= \boldsymbol{\varepsilon}^\top\boldsymbol{R_p}^\top(\dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_{\boldsymbol{p}})\end{aligned} \tag{3.66}$$

At this point, the kinematic and dynamic parts of the task are considered to be solved. This means that the velocity of the tracking point $\dot{\boldsymbol{p}}$ is equal to a desired virtual velocity vector $\dot{\boldsymbol{p}}_{dv}$. Hence $\dot{V}_\varepsilon$ becomes:

$$\dot{V}_\varepsilon = \boldsymbol{\varepsilon}^\top\boldsymbol{R_p}^\top(\dot{\boldsymbol{p}}_{dv} - \dot{\boldsymbol{p}}_{\boldsymbol{p}}) \tag{3.67}$$

The desired velocity vector and the velocity of the virtual particle are then designed to respect the stability criterion $\dot{V}_\varepsilon \leq 0$.

In [20], the desired virtual velocity vector $\dot{\boldsymbol{p}}_{dv}$ is constructed so as to introduce LOS

guidance. Here, according to the results of section 3.5.2, the desired velocity vector can be chosen as the output of the position control law $\boldsymbol{\lambda}_1$ where the parameter α is chosen as the position of the virtual particle $\boldsymbol{p_p}$. Using the $\mathcal{H}$ based controller, one gets $\dot{\boldsymbol{p}} = \dot{\boldsymbol{\eta}}_{\boldsymbol{E}1} \equiv \boldsymbol{\lambda}_1(\boldsymbol{\eta_E}, \dot{\boldsymbol{\eta}}_{\boldsymbol{E}}, \boldsymbol{p_p}) = \dot{\boldsymbol{p}}_{dv}$.

In a first approximation, the control law $\boldsymbol{\lambda}_1$ is chosen as a proportional controller based on the following error $\boldsymbol{\varepsilon}$ and moved to the inertial frame using $\boldsymbol{R_p}$ associated with a feedforward term:

$$\boldsymbol{\lambda}_1 = \boldsymbol{R_p v_d} - \boldsymbol{\Lambda R_p \varepsilon} \tag{3.68}$$

with $\boldsymbol{\Lambda}$ a diagonal gain matrix with equal coefficients on the diagonal $\boldsymbol{\Lambda} = \lambda \mathbb{I}_3$ with $\lambda > 0$. The vector $\boldsymbol{v_d} = [U_d, 0, 0]^\top$ is the desired velocity of the vehicle expressed in the path frame where the desired surge speed U_d is a control parameter defined in the following.

Then, because the virtual path particle is constrained to the path, its velocity vector in the inertial frame is defined as:

$$\dot{\boldsymbol{p}}_{\boldsymbol{p}} = \boldsymbol{R_p v_p} \tag{3.69}$$

with $\boldsymbol{v_p} = [U_p, 0, 0]^\top$ and U_p the speed of the particle which is a control parameter, chosen later to ensure stability and move the virtual particle according to the behavior of the vehicle.

The time derivative of the Lyapunov function candidate becomes:

$$\begin{aligned} \dot{V}_{\varepsilon} &= \boldsymbol{\varepsilon}^\top \boldsymbol{R_p}^\top (\boldsymbol{R_p v_d} - \boldsymbol{\Lambda R_p \varepsilon} - \boldsymbol{R_p v_p}) \\ &= \boldsymbol{\varepsilon}^\top (\boldsymbol{v_d} - \boldsymbol{\Lambda \varepsilon} - \boldsymbol{v_p}) \\ &= -\lambda(x_e^2 + y_e^2 + z_e^2) + (U_d - U_p) x_e \end{aligned} \tag{3.70}$$

Here, the speed of the virtual particle U_p can be chosen to cancel the second term of the right hand member, $U_p = U_d$ and thus $\dot{V}_{\varepsilon} \leq 0$ for all $\lambda > 0$.

The path-following problem can be updated to a *path-tracking* problem as described in [20]. In path-tracking, the speed of the vehicle is adapted to catch up with the particle. To do so, the desired speed of the vehicle is expressed as a function of the along-track error:

$$U_d = U_p - k_x x_e \tag{3.71}$$

with k_x a gain parameter.

The time derivative of the Lyapunov function candidate becomes:

$$\dot{V}_{\boldsymbol{\varepsilon}} = -\lambda(x_e^2 + y_e^2 + z_e^2) - k_x x_e^2 \tag{3.72}$$

In such a case, the equilibrium $\boldsymbol{\varepsilon} = \mathbf{o}$ is stable for any $\lambda > 0$ and $k_x > 0$.

This last choice of U_d is particularly interesting since it allows an operator to choose the behavior of the whole system (vehicle and path particle) by tuning only the particle speed U_p and the control parameters. Overall, the equilibrium $\boldsymbol{\varepsilon} = \mathbf{o}$ is shown to be globally exponentially stable.

The path tracking scheme can easily be adapted to other scenarios. As an example, in a robotic fleet, the path particle can represent a primary vehicle following a predefined task. Other secondary vehicles of the fleet are tracking the primary one for instance going back and forth to the surface for communication. It is then very important that all the secondary vehicles adapt their speed to the primary one.

Another similar solution to equation (3.71) is to choose the desired speed of the vehicle and adapt the particle speed so that the vehicle is not left behind. This is a different approach where the desired speed of the ship is prioritized. It implies that no time constraint can be added to the task as the behavior of the vehicle may vary between the waypoints. In this case, the particle speed is:

$$U_p = U_d + k_x x_e \tag{3.73}$$

The derivative of the Lyapunov function is obviously similar to equation (3.72) using this solution. One major difference is that the path particle speed U_p may not meet with assumptions **A.1** and **A.2**. This approach also creates the stable equilibrium $\boldsymbol{\varepsilon} = \mathbf{o}$ for any $\lambda > 0$ and $k_x > 0$. Yet, it does not ensures that the path is regularly defined. The main advantage is that it allows a pilot to choose the desired speed of the craft and the particle adapts.

3.5.4 Simulation results

The controller developed in Sections 3.5.2 and 3.5.3 is applied in simulation to the model of the Remus100. These simulations were performed on a different simulator based on the MSS toolbox [70]. Experiments are conducted with a simulated measurement noise of standard deviation $0.1\,\mathrm{m}$ for the position measurements and 0.01rad for

the orientations. The path is defined as straight lines in 3-D joining waypoints. No additional constraint is added to the task and the vehicle is naturally stable in roll and pitch. Note that the shape and actuation configuration of the Remus100 allows neglecting the sway and heave forces generated by the actuators in the controller. The vehicle is considered actuated in surge, pitch and yaw.

A simplified allocation procedure is used to calculate the defection angles of the rudder and stern plane of the vehicle δ_r and δ_s:

$$\delta_r = \frac{2N_c}{\frac{L}{2}\rho U_h^2 A_s C_{\delta_s}} \tag{3.74a}$$

$$\delta_s = \frac{2M_c}{\frac{L}{2}\rho U_v^2 A_r C_{\delta_r}} \tag{3.74b}$$

where M_c and N_c are the fifth and sixth components of the control vector τ_c, ρ is the local water density, L is the length of the hull of the vehicle, U_h and U_v are the $\boldsymbol{y_B}$ axis and $\boldsymbol{z_B}$ axis relative speeds of the vehicle and A_s, A_r, C_{δ_s} and C_{δ_r} are known physical parameters of the control surfaces.

The target switches from one waypoint to the next as soon as the tracking point of the vehicle arrives in a circle of $0.1\,\mathrm{m}$ radius around the current target.

3.5.4.1 Without external disturbance

With no external current, the controller gives very good following results. The position of the tracking point and virtual particle are shown on figure 3.16. Figure 3.17 shows that the vehicle lines up with the path between the turns and stays tangent to the x axis of the path frame at all times. The pitch and yaw angles follow the desired elevation and azimuth angles (respectively) of the path.

3.5.4.2 With sea current

In this experiment, the vehicle is exposed to a $0.5\,\mathrm{m\,s^{-1}}$ sea current aligned with the $\boldsymbol{y}_\mathbf{o}$ axis. Figure 3.18 shows that the following per shows that the path following performances are not decreased in presence of external disturbance and measurement noise. Nonetheless, figure 3.19 shows that the yaw equilibrium taken by the vehicle between the turns is different from the azimuth angle of the path. With sideways current, the

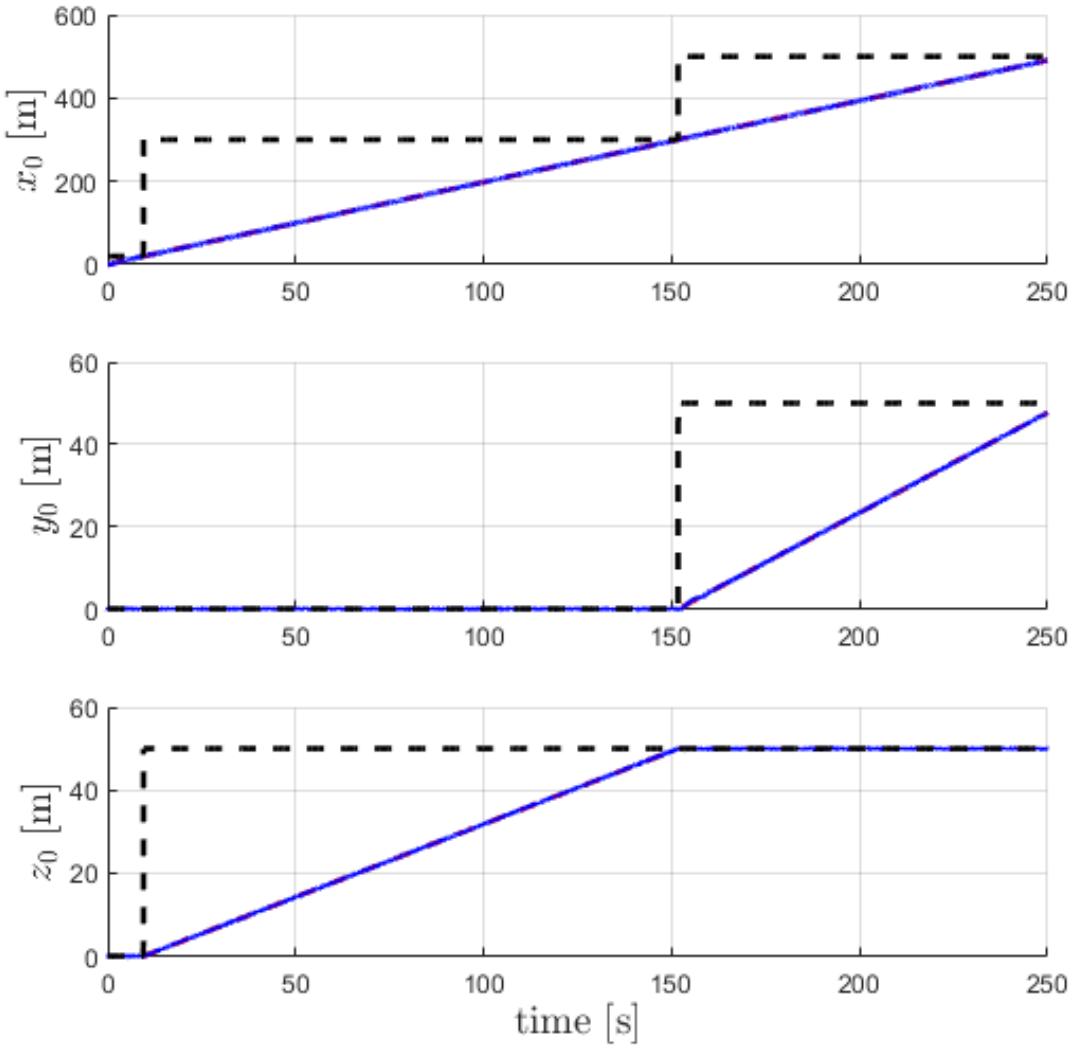


Figure 3.16 – Position of the tracking point (Blue) of the vehicle with no current. Dashed Red: Position of virtual particle, Dashed black: Waypoints

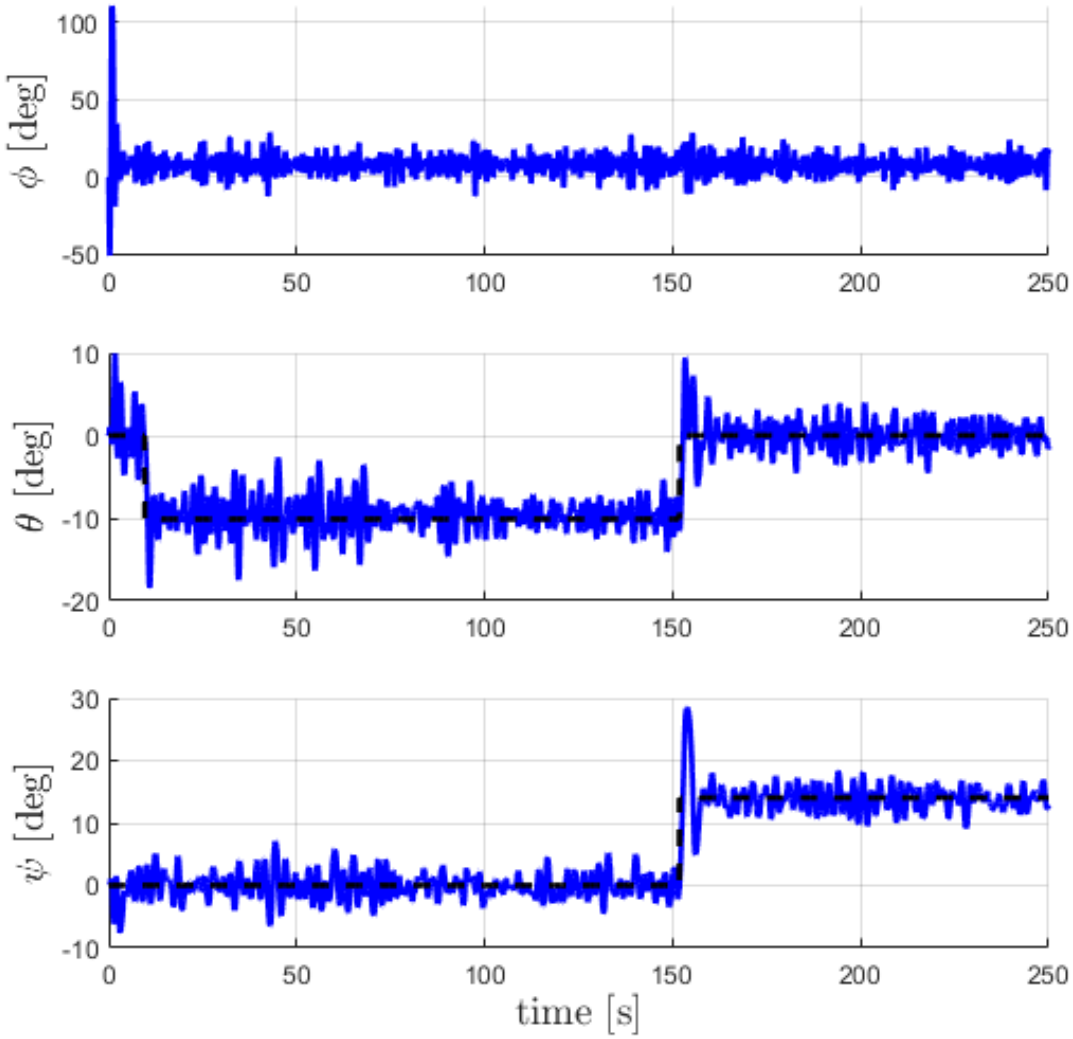


Figure 3.17 – Attitude of the Remus100 vehicle (Blue) compared to the azimuth and elevation angle of the path (Dashed black)

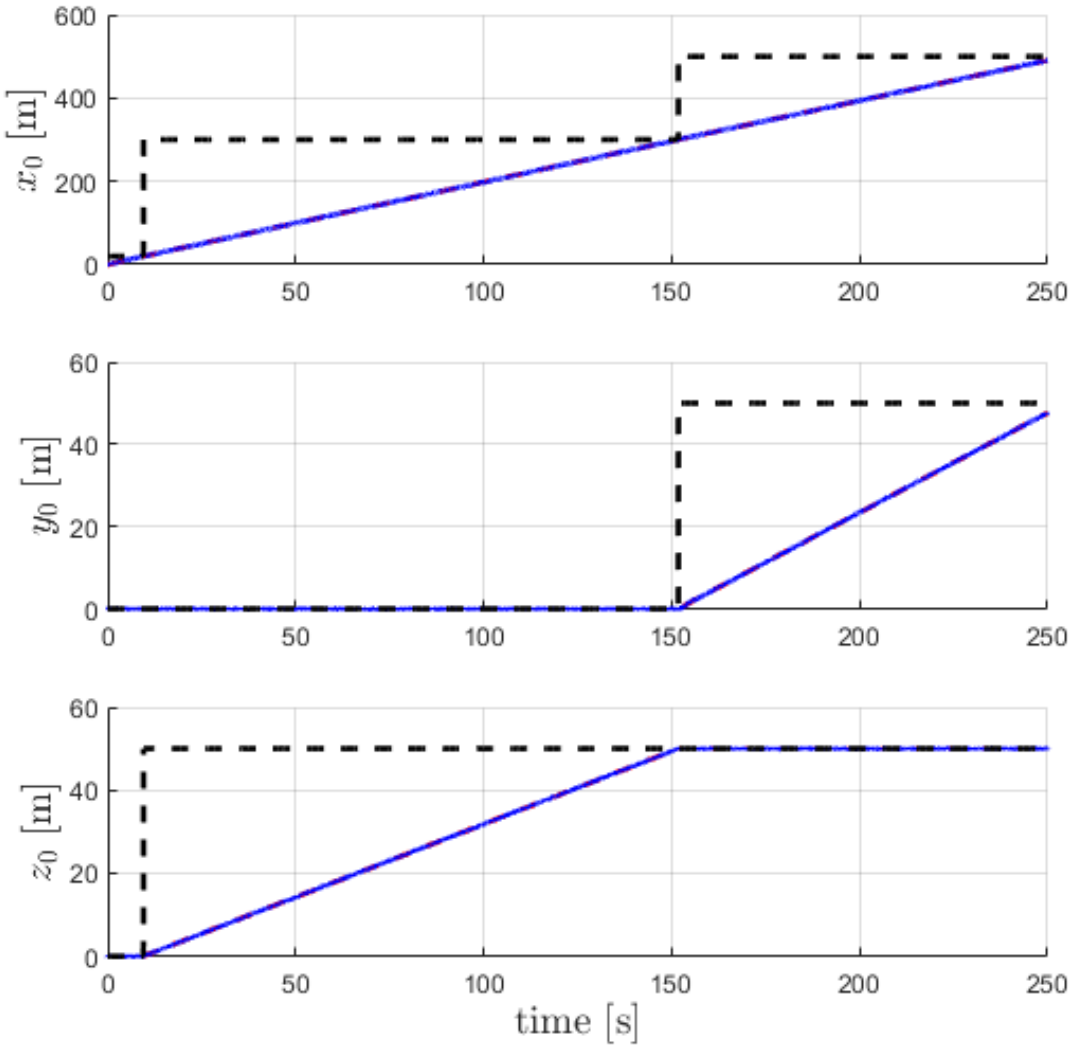


Figure 3.18 – Position of the tracking point (Blue) of the vehicle with current. Dashed Red: Position of virtual particle, Dashed black: Waypoints

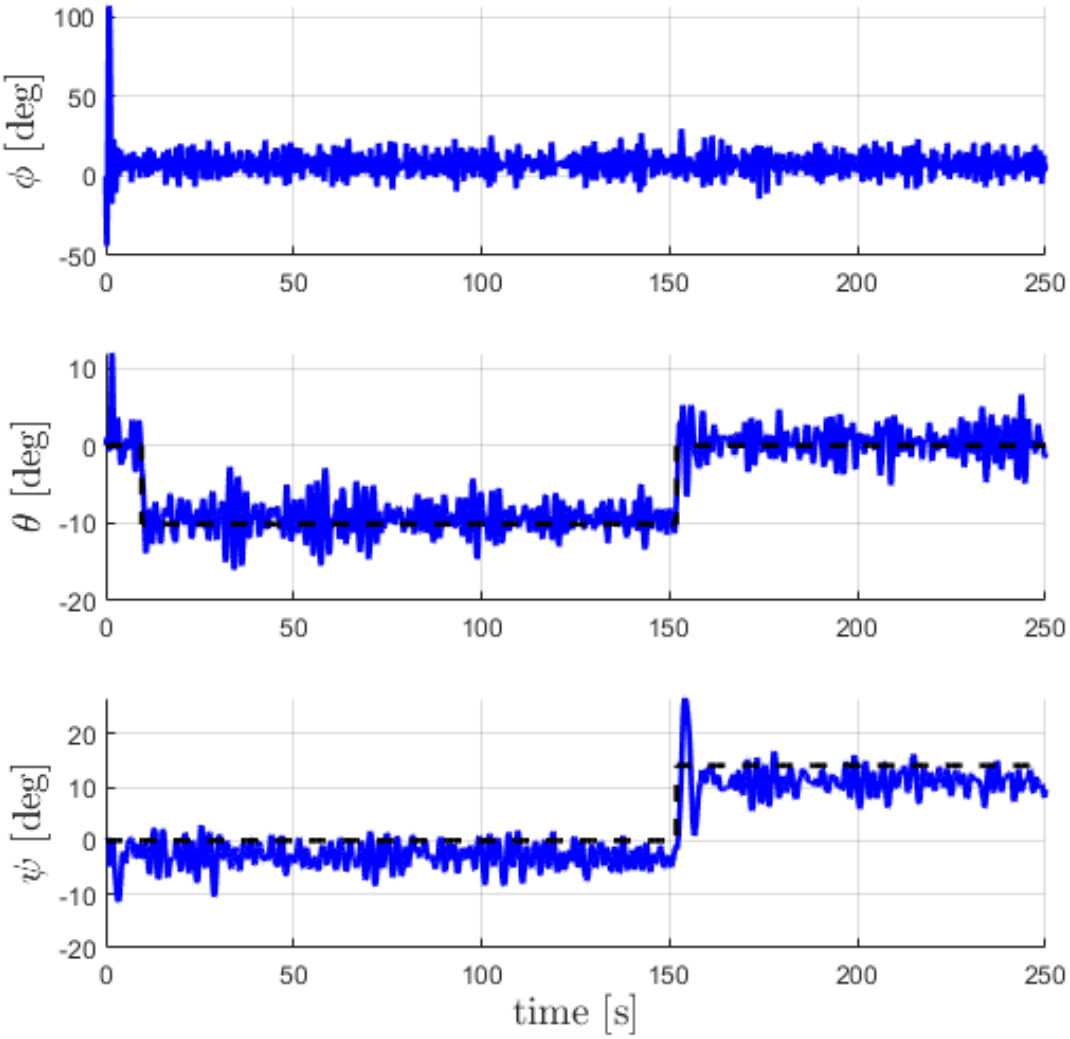


Figure 3.19 – Attitude of the Remus100 vehicle (Blue) with sea current compared to the azimuth and elevation angle of the path (Dashed black)

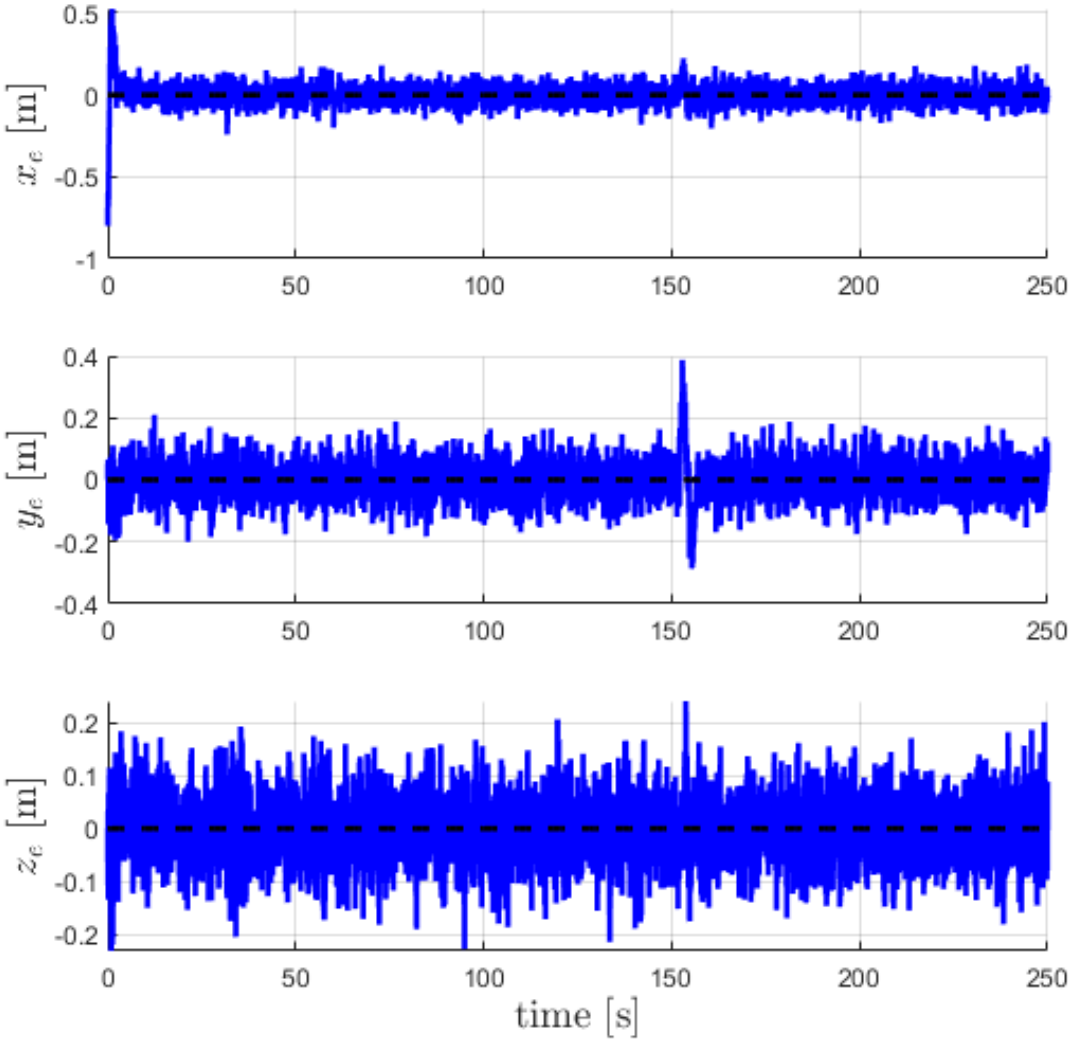


Figure 3.20 – Path following error in the path frame.

vehicle cannot stay tangent to the path but finds a different equilibrium. Yet, figure 3.20 shows that the error in the path frame is canceled even with external disturbance. Even though robustness to external disturbance has not been demonstrated on this example, the cascade structure of the controller, integral action and use of a tracking point hint towards robustness to bounded disturbances.

3.6 Partial conclusions on $\mathcal{H}$ based control

In this chapter, a novel control method for underactuated marine craft has been presented. This controller relies on a guidance stage based on the new Handy matrix $\mathcal{H}$. The matrix creates compensation behaviors between the different DOF of the system. Notably, it allows exploiting actuated rotations of the vehicle to compensate for the lack of some translations as it is usually the case with marine vehicles. The method has been shown to perform as well as a Line Of Sight guidance principle and a Sliding Mode Controller, two reference control methods in the literature.

The convergence calculations and simulation results show that the errors on the controlled DOF globally exponentially converge to zero. The cascade structure of the system makes it reliable and robust to external disturbances. The $\mathcal{H}$ matrix does not create any pole or zero in the control law which entirely characterized by the gains of proportional controller of the inner loop and the PI of the outer loop.

Nevertheless, this controller shows one of the drawbacks of any compensation control methods. Because the attitude taken by the vehicle during the task is not completely controlled, and notably in the examples above, the roll angle is not controlled, the vehicle can stabilize in configurations in which actuation over one or several DOF of task is lost. It is notably showing clearly on section 3.4.1. If the vehicle were to stabilize with a roll angle of $90°$, actuation over the rotation around the vertical axis of the inertial frame z_o would be lost and the vehicle would not be able to meet the task constraints. The equilibrium on the DOF used for compensation, roll in the examples, is hardly predictable as it depends on AUV dynamics and environmental factors. As a matter of facts, limitations can be set on the DOF used for compensation to avoid singular attitudes with the counterpart of reducing performance.

The kinematic guidance principle has also been applied to the path-following problem. This problem and the Remus100 vehicle used for example are a very common underwater robotics problem. The $\mathcal{H}$ matrix controller allows perfect path-following in this

case even with unmodeled disturbances and measurement noise. This path-following problem could easily be extended to a leader-follower fleet problem where one vehicle as to follow another one performing a specified mission.

Further studies on physical systems will be conducted to discriminate the different compensation solutions offered by the method. Notably, an energetic study comparing the total energy consumption of two strategy would allow choosing one optimal solution over the other and help for the choice of a solution.

CHAPTER 4

FLATNESS-BASED CONTROL OF MARINE VEHICLES

Contents

In this chapter, flatness-based controllers are derived for both the fully actuated marine craft and the underactuated surface vessel. Flatness-based control has been briefly introduced in section 2.3. It has shown very good results on both fully actuated and under actuated systems as the mechanical crane or the car with trailers. The main quality of flatness-based control is its robustness to both model approximations and disturbances making it a strong contestant for the control of marine craft.

As shown in this chapter, flatness-based controller relies on the inherent properties of a physical system. This chapter demonstrates that the flat character of a system can also depend on the actual numerical values of some sets of model parameters. This last property is exploited in this chapter to derive a flatness-based controller for systems which are not naturally flat.

To further increase the robustness of flatness-based control on udrcatutaed vehicles, it is associated with Intellignt PID section 2.5.1. Here, the iPID is used to allow applying the flatness based controller calculated on a special case of surface vessels, the Hovercraft, to any surface vessel. The iPID also compensates one of the weaknesses of flatness-based control, its unstable behavior away from the trajectory.

4.1 Introduction to flatness based control

In this section, the concept of *flatness-based* control is introduced with a theoretical example derived from [90] and studied in details in [84]. It is a simple MIMO system given as:

$$\dot{x}_1 = u_1 \tag{4.1a}$$

$$\dot{x}_2 = x_3 + x_2 u_1 \tag{4.1b}$$

$$\dot{x}_3 = u_2 \tag{4.1c}$$

where u_1 and u_2 are the two control inputs and $\boldsymbol{x}[x_1, x_2, x_3]^\top$ is the state.

As evoked before, *(differential) flatness*, is an inherent property of a class of systems, notably useful for the design of controllers and often associated with feedforward linearization and extended PID control [84]. First, the plant must be shown *flat* meaning that all variables of the system can be expressed as function of a particular set of variables, the *flat outputs*, and their derivatives. The flat output is a set of independent differential functions of the system variables which can be used to express all other variables of the system [139]. The flat output must have as many components as the number of inputs of the plant. A flat system can have several flat outputs.

In this example, the system has two inputs. The flat output must therefore have two independent components. On physical systems, the flat output is often chosen as a set of variables of interest for the system. Working with mobile robots for instance, the position and orientation of the vehicle are good candidates for the flat output. Also, the

task is likely to be defined in terms of desired values of the flat output making a strong argument in the choice of the flat output when several solutions exist.

To show flatness of a system, flat output candidates are tested. The first flat output candidate denoted $\boldsymbol{z}$ is chosen as: $\boldsymbol{z} = [x_1\ x_2]^\top$ in this example. Showing that this system is flat means demonstrating that all other variables of the system can be expressed as functions of the flat output and its derivatives without solving any differential equation.

The three variables of the system are therefore expressed as functions of x_1 and x_2:

$$u_1 = \dot{x}_1 \tag{4.2a}$$

$$x_3 = \dot{x}_2 - x_2\dot{x}_1 \tag{4.2b}$$

$$u_2 = \ddot{x}_2 - \dot{x}_2\dot{x}_1 - x_2\ddot{x}_1 \tag{4.2c}$$

These formulations of the intermediary variable x_3 and the two inputs u_1 and u_2 show that the system is flat. The flat output candidate $\boldsymbol{z}$ is indeed a flat output of this system. Equations like (4.2) will be referred to as *equations of flatness* or *flatness equations* in the following. They are notably used as a base in the control calculations.

In this example, the pair x_1, x_2 is the most natural flat output. Whereas some systems may accept several flat outputs, here equation (4.1b) implies that any candidate containing x_3 would require integrating a differential equation to show flatness. For instance, if $\boldsymbol{z}' = [x_1\ x_3]^\top$ were to be chosen, x_2 could not be expressed as an algebraic function of the flat output but instead as a differential expression $\dot{x_2} = x_3 + x_2\dot{x_1}$ preventing demonstration of flatness. This is referred to as a flatness *defect* (see [53] for more details).

One important property of flat systems, stated for instance in [38], is that all flat systems can be represented using the *Brunovský state* [23]. The Brunovský representation of system is a minimal representation where a succession of change of variables allow representing the system as a set of integrator chains. The Brunovský state representation is particularly useful for the design of the control law in flatness-based control as it allows turning complex non-linear systems into simpler integrator chains. Note that [38] introduces a *controller form* based on the Brunovský state. A simple feedback takes the system from the controller form to the Brunovský form.

Quoting [84] again, the Brunovský state for this system is:

$$\xi_{1,1} = x_1 \tag{4.3a}$$
$$\xi_{2,1} = x_2 \tag{4.3b}$$
$$\xi_{2,2} = \dot{x}_2 \tag{4.3c}$$

Then, the system can be rewritten as:

$$\dot{\xi}_{1,1} = u_1 \tag{4.4a}$$
$$\dot{\xi}_{2,1} = \xi_{2,2} \tag{4.4b}$$
$$\dot{\xi}_{2,2} = \xi_{2,2}u_1 + \xi_{2,1}\dot{u}_1 + u_2 \tag{4.4c}$$

At this point, introducing the new inputs $v_1 = u_1$ and $v_2 = \xi_{2,2}u_1 + \xi_{2,1}\dot{u}_1 + u_2$ allows writing equation (4.4) as two simple chains of integrators, which facilitates the calculations of the controller and leads to an unique solution for the original inputs:

$$u_1 = v_1 \tag{4.5a}$$
$$u_2 = v_2 - \xi_{2,2}\dot{\xi}_{1,1} - \xi_{2,1}\ddot{\xi}_{1,1} \tag{4.5b}$$

The new system reads:

$$\dot{\xi}_{1,1} = v_1 \tag{4.6a}$$
$$\dot{\xi}_{2,1} = \xi_{2,2} \tag{4.6b}$$
$$\dot{\xi}_{2,2} = v_2 \tag{4.6c}$$

The control inputs for this new simplified system are obviously calculated in a feedforward and extended PID fashion as:

$$v_1^c = \dot{\xi}_{1,1}^* + \text{PI}(e_{1,1}) \tag{4.7a}$$
$$v_2^c = \dot{\xi}_{2,2}^* + \text{PID}(e_{2,1}) \tag{4.7b}$$

In equation (4.7), the upperscript $*$ denotes desired values and the functions $\text{PI}(\xi_1)$ and $\text{PID}(\xi_2)$ represent usual PI and PID controllers calculated on the errors on $\xi_{1,1}$ and $\xi_{2,1}$ respectively.

Using equation (4.5), the original control inputs are calculated as:

$$u_1^c = \dot{\xi}_{1,1}^* + \text{PI}(\xi_{1,1}) \tag{4.8a}$$
$$u_2^c = \dot{\xi}_{2,2}^* + \text{PID}(\xi_{2,1}) - \xi_{2,2}^*\dot{\xi}_{1,1}^* - \xi_{2,1}^*\ddot{\xi}_{1,1}^* \tag{4.8b}$$

where all variables of lesser order have been replaced by their desired values issued from the trajectory. Because all the variables of the system have been defined as functions of the flat output, the desired values for each of them can be calculated for the complete trajectory. Using the desired values of the system variables in the controller gives flatness-based control its robustness but also limits the range of application of flatness-based control to the vicinity of the trajectory. This highlights that the trajectory of the flat output must be sufficiently smooth and continuously defined up to a high enough order, here to the second order, to allow proper feedforward linearization.

In facts, a faster but less rigorous method —working on most flat systems— can be used to build the controller without writing the Brunovský state. The idea is to seek, in each flatness equation of inputs (in this example equation (4.2a) and (4.2c)), a *favored* or *preferential* flat output component and replace it with an extended PID controller and a feedforward term of the higher order. The rest of the variables are replaced by their desired values in a feedforward linearizing manner. The preferential output component generaly appears with a higher derivation order in the input flatness equation. Here, the choice is obvious in equation (4.2): the first input u_1 only depends on the first order derivative of the first input component $\dot{x}_1$ so u_1 can only be used to control x_1. For u_2, both x_1 and x_2 appear at the second order in equation (4.2c) but x_1 has already been chosen in u_1 so, to avoid redundancies, only x_2 can be used. Following this rule, the two control inputs are:

$$u_1^c = \dot{x}_1^* + \text{PI}(e_1) \tag{4.9a}$$
$$u_2^c = \ddot{x}_2^* + \text{PID}(e_2) - \dot{x}_2^*\dot{x}_1^* - x_2^*\ddot{x}_1^* \tag{4.9b}$$

Note that, because x_1 appears at the first order in u_1 a PI controller is sufficient while in u_2, because x_2 appears at the second order, a PID controller is better suited [84, 85]. In this example, the resulting expression (4.9) is equivalent to equation (4.8).

This method does not always allow designing the controller for flat systems and would not work on a system where no flat output component appear preferential in any of the input flatness equations (see for instance the hovercraft system in section 4.3.3).

4.2 Flatness-based control in the fully actuated case

In this section, flatness-based control is applied to the example of a fully-actuated underwater vehicle (see for example the vehicle introduced in section 1.2.1.1). On fully-actuated plants, the end result of flatness-based control is naturally close to feedforward linearizing control but the approach is different. One of the few applications of flatness to an underwater vehicle is found in [137] but a variation of the method is used here.

The model of the fully-actuated underwater vehicle is recalled here. It is expressed in the inertial frame [8] as it is the most self-explanatory representation for flatness-based control but the same results are obtained with the dynamic model expressed in the mobile frame as in equation (1.5).

$$\boldsymbol{\tau} = \tilde{\boldsymbol{M}}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \tilde{\boldsymbol{C}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{D}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{4.10a}$$

$$\tilde{\boldsymbol{M}}(\boldsymbol{\eta}) = \boldsymbol{M}\boldsymbol{J}(\boldsymbol{\eta})^{-1} \tag{4.10b}$$

$$\tilde{\boldsymbol{C}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}) = \boldsymbol{M}\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{J}(\boldsymbol{\eta})^{-1} \tag{4.10c}$$

$$\tilde{\boldsymbol{D}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}) = \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{J}(\boldsymbol{\eta})^{-1} \tag{4.10d}$$

where $\tilde{\boldsymbol{M}}(\boldsymbol{\eta})$, $\tilde{\boldsymbol{C}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})$ and $\tilde{\boldsymbol{D}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})$ are the model matrices expressed in the inertial frame (see section 1.1.3 for details). In the following, the arguments of matrices $\tilde{\boldsymbol{M}}$, $\tilde{\boldsymbol{C}}$ and $\tilde{\boldsymbol{D}}$ will be omitted for the sake of clarity and conciseness.

In this chapter, as in many of the examples of chapter 2, the vector of propulsive forces and moments $\boldsymbol{\tau}$ is considered to be the input of the system. In facts, when working with fixed thrusters, the real input of the system is the vector of thrusts $\boldsymbol{u}$. Yet, the relationship $\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{u}$ being static, using $\boldsymbol{\tau}$ as an input is acceptable with fixed thrusters. With reconfigurable thrusters or control surfaces, the relation between $\boldsymbol{\tau}$ and the real inputs of the system (like the rudder angle or the reconfiguration angle of a vector thruster) may prevent showing flatness of the complete system as it adds more non linearities in the model.

4.2.1 Flat Output

From a broader point of view, the flat output can be seen as a set of variables of interest of the system whom main property is that it is naturally *central* to the system

in the sense that all other variables of the system can be expressed as functions of this set and its time derivatives. The relationship between the flat output and the other variables is inherent to the system. In the fully actuated case the choice of the flat output candidate comes naturally as $\boldsymbol{z} = \boldsymbol{\eta}$.

A different choice like $\boldsymbol{z} = \boldsymbol{\nu}$ could work but would be more complex to use. It would for instance be harder to have order zero position terms in the controller or to add an integral effect on the position and orientation.

4.2.2 Flatness of the system

In the fully actuated example shown in equation (4.10), the only variables other than the flat output candidate $\boldsymbol{z} = \boldsymbol{\eta}$ are the input vector $\boldsymbol{\tau}$ and the velocity vector in the mobile frame $\boldsymbol{\nu}$. The dynamic model equation (4.10) of the example and the kinematic model equation (1.1) give naturally:

$$\boldsymbol{\nu} = \boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\boldsymbol{\eta}} \tag{4.11a}$$

$$\boldsymbol{\tau} = \tilde{\boldsymbol{M}}\ddot{\boldsymbol{\eta}} + \tilde{\boldsymbol{C}}\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{D}}\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{4.11b}$$

The fully actuated underwater vehicle is then flat with the flat output $\boldsymbol{z} = \boldsymbol{\eta}$. In this case, the equations of the model directly give the equations of flatness but it will not always be the case and notably not with underactuated systems.

4.2.3 Description of the trajectory

The task trajectory is usually defined in terms of desired values of the flat output. It must be *sufficiently smooth* meaning that it must be defined and differentiable to a sufficiently high order to match the order of derivation of the flat output needed to defined all variables of the system.

One methodology to define a suitable trajectory to flatness control is to start with a set of waypoints. Each waypoint imposes a desired position, velocity and acceleration on all six DOF of the flat output as well as a time constraint. The order of constraint of the waypoints must match the order of the system. Here in the case of the AUV, the trajectory must be defined up to the second order. Then, smooth paths are defined between the waypoints using, for instance, a piece-wise polynomial trajectory generator. It

is mandatory to keep continuity around the waypoints between two consecutive pieces of path up to the highest order of the system.

Once the desired values of the flat output are properly defined, nominal values of the other variables of the system can be calculated using the flatness equations. Notably, a nominal input $\boldsymbol{\tau}^*$ is calculated:

$$\boldsymbol{\nu}^* = \boldsymbol{J}(\boldsymbol{\eta}^*)^{-1}\dot{\boldsymbol{\eta}}^* \tag{4.12a}$$

$$\boldsymbol{\tau}^* = \tilde{\boldsymbol{M}}(\boldsymbol{\eta}^*)\ddot{\boldsymbol{\eta}}^* + \tilde{\boldsymbol{C}}(\boldsymbol{\eta}^*, \dot{\boldsymbol{\eta}}^*)\dot{\boldsymbol{\eta}}^* + \tilde{\boldsymbol{D}}(\boldsymbol{\eta}^*, \dot{\boldsymbol{\eta}}^*)\dot{\boldsymbol{\eta}}^* + \boldsymbol{g}(\boldsymbol{\eta}^*) \tag{4.12b}$$

4.2.4 Flatness-based controller

As briefly evoked in section 2.2.1, flatness-based controllers in the fully-actuated case often resemble a sort of feedforward linearizing controller. In facts, the nominal input defined with respect to the flat output equation (4.12b) provides feedforward terms that are used in the flatness-based controller. The closed loop system obtained with a flatness-based controller is nonlinear but the nonlinearities cancel out in the vicinity of the trajectory. Flatness-based controllers are therefore particularly suited for applications where the system is assured to stay in the neighborhood of the desired trajectory. As shown in [85], the remaining non-linear terms of the closed-loop system can behave favorably for the convergence and stability of the system.

The rigorous method to designing the flatness-based controller of a system starts with writing the model in the Brunovský state as shown in section 4.1. Yet, writing the Brunovský state representation for such a system can be quite strenuous as it requires separating and decoupling all six DOF of the flat output. Instead, the other method is used in this example. In facts, because the flat output and the input of the system are defined in two different frames related by the transformation matrix $\boldsymbol{J}(\boldsymbol{\eta})$, no favored output component arise in any of the input flatness equations. The matrix $\tilde{\boldsymbol{M}}(\boldsymbol{\eta})$ is non-diagonal for any vector $\boldsymbol{\eta}$ thus several components of the flat output appear at the second order on all the input equations. Yet, provided that the $\tilde{\boldsymbol{M}}$ matrix is of plain rank at all times, the vector formulation equation (4.11b) can be used and the complete $\ddot{\boldsymbol{\eta}}$ vector replaced with the feedforward term and the controller. This condition on $\tilde{\boldsymbol{M}}$ ensures that, for any configuration of the vehicle, no output component will be left uncontrolled during the application and no redundancy can appear.

The controller is designed directly on the vector flatness formulation equation (4.11b) replacing the second order instance of the complete flat output $\ddot{\boldsymbol{\eta}}$ by a vector of six decoupled PID controllers, one per flat output, and a feedforward term $\ddot{\boldsymbol{\eta}}^*$. The other variables are substituted with their desired values. The controller reads:

$$\boldsymbol{\tau}_c = \tilde{\boldsymbol{M}}(\boldsymbol{\eta}^*)(\ddot{\boldsymbol{\eta}}^* + \text{PID}(\boldsymbol{e}_\eta)) + \tilde{\boldsymbol{C}}(\boldsymbol{\eta}^*, \boldsymbol{\nu}^*)\dot{\boldsymbol{\eta}}^* + \tilde{\boldsymbol{D}}(\boldsymbol{\eta}^*, \boldsymbol{\nu}^*)\dot{\boldsymbol{\eta}}^* + \boldsymbol{g}(\boldsymbol{\eta}^*) \tag{4.13}$$

In this equation, $\text{PID}(\boldsymbol{e}_\eta)$ is a vector-PID controller calculated with the complete error vector $\boldsymbol{e}_\eta$. The gain matrices are diagonal. It is equivalent to having six independent PID controllers, one for each component of the flat output.

Applying this controller to the system (4.10) leads to the following closed-loop system:

$$\begin{aligned}\tilde{\boldsymbol{M}}(\boldsymbol{\eta}^*)(\ddot{\boldsymbol{\eta}}^* + \text{PID}(\boldsymbol{e}_\eta)) - \tilde{\boldsymbol{M}}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \tilde{\boldsymbol{C}}(\boldsymbol{\eta}^*, \boldsymbol{\nu}^*)\dot{\boldsymbol{\eta}}^* - \tilde{\boldsymbol{C}}(\boldsymbol{\eta}, \boldsymbol{\nu})\dot{\boldsymbol{\eta}} \\ + \tilde{\boldsymbol{D}}(\boldsymbol{\eta}^*, \boldsymbol{\nu}^*)\dot{\boldsymbol{\eta}}^* - \tilde{\boldsymbol{D}}(\boldsymbol{\eta}, \boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}^*) - \boldsymbol{g}(\boldsymbol{\eta}) = 0\end{aligned} \tag{4.14}$$

The resulting closed-loop system is nonlinear because of the feedforward terms used for linearization. Yet, in the neighborhood of the trajectory, the non linearities cancel out and, as demonstrated in [85], the non linearities of the closed-loop system can behave favorably. Notably, when the model parameters are approximated in the control calculations, the remaining non-linear terms can add robustness to the controller. The performances of this flatness-based controller are detailed and compared with other controllers in section 4.5.1.

4.3 Flatness-based control in the underactuated case

As seen with several examples throughout this work (see chapter 2) a large number of marine vehicles are underactuated or ill-actuated. This is notably the case of most surface ships. While flatness-based control is not typically associated to underactuated vehicles, some examples outside of the marine context show good control performances on underactuated systems. Flatness-based control has notably made a name for itself solving the problem of controlling a car with trailers [53]. Also, the natural robustness to external disturbance and model approximations of flatness-based control makes it an ideal candidate for marine applications.

4.3.1 Model of the underactuated surface vessel

The underactuated surface vessel has been used as an example in several studies on flatness-based control [154, 135]. In [154], two different models are presented for the surface ship giving similar results. In this section, the model issued from [63] for surface vessels is used. Reminding the equations introduced in chapter 1 the model matrices are given for the surface vessel considering that the body-fixed frame is centered on the center of gravity as:

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & I_0 - N_{\dot{r}} \end{bmatrix} \tag{4.15a}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -(m - Y_{\dot{v}})v + Y_{\dot{r}}r \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v - Y_{\dot{r}}r & -(m - X_{\dot{u}})u & 0 \end{bmatrix} \tag{4.15b}$$

$$\boldsymbol{D}(\boldsymbol{\nu}) = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \tag{4.15c}$$

Hence, the complete model of a surface vehicle actuated in surge and yaw (neglecting possible coupled sway actuation force) is given in component form as:

$$\dot{x} = u\cos\psi - v\sin\psi \tag{4.16a}$$
$$\dot{y} = u\sin\psi + v\cos\psi \tag{4.16b}$$
$$\dot{\psi} = r \tag{4.16c}$$
$$\dot{u} = \frac{1}{m - X_{\dot{u}}}\left(\tau_u + (m - Y_{\dot{v}})vr - Y_{\dot{r}}r^2 + X_u u\right) \tag{4.16d}$$
$$\begin{aligned}\dot{v} = &\frac{I_0 - N_{\dot{r}}}{(m - Y_{\dot{v}})(I_0 - N_{\dot{r}}) - Y_{\dot{r}}^2}\left(Y_{\dot{r}}\dot{r} - (m - X_{\dot{u}})ur + Y_v v + Y_r r\right)\\ &- \frac{Y_{\dot{r}}}{(m - Y_{\dot{v}})(I_0 - N_{\dot{r}}) - Y_{\dot{r}}^2}\left(\tau_r + N_{\dot{v}}\dot{v} - (X_{\dot{u}} - Y_{\dot{v}})uv + Y_{\dot{r}}ur + N_v v + N_r r\right)\end{aligned} \tag{4.16e}$$
$$\begin{aligned}\dot{r} = &-\frac{Y_{\dot{r}}}{(m - Y_{\dot{v}})(I_0 - N_{\dot{r}}) - Y_{\dot{r}}^2}\left(Y_{\dot{r}}\dot{r} - (m - X_{\dot{u}})ur + Y_v v + Y_r r\right)\\ &+ \frac{m - Y_{\dot{v}}}{(m - Y_{\dot{v}})(I_0 - N_{\dot{r}}) - Y_{\dot{r}}^2}\left(\tau_r + N_{\dot{v}}\dot{v} - (X_{\dot{u}} - Y_{\dot{v}})uv + Y_{\dot{r}}ur + N_v v + N_r r\right)\end{aligned} \tag{4.16f}$$

Equation (4.16) gives the kinematics and dynamics of an underactuated ship in the horizontal plane using the usual approximations of shape, added masses and damping relative to surface vessels. The model is expressed at the center of the mobile frame which is usually chosen at the center of gravity of the vehicle in surface applications. The two inputs of this system are τ_u and τ_r, respectively a surge force and a yaw moment generated by either two fixed thrusters, a single vector thruster or a thruster associated with a control surface (in the two last cases, sway disturbances are neglected).

The non diagonal shape of the added mass and damping matrices create several coupled acceleration and speed terms in equation (4.16). Notably, a quadratic yaw speed term $Y_{\dot{v}}r^2$ appears in the surge dynamics equation (4.16d) and a yaw acceleration term $Y_{\dot{r}}\dot{r}$ appears in the sway dynamics equation (4.16e). These terms are often neglected as they will be in what follows.

4.3.2 Flatness defects of the surface vessel

Considering that most applications of the autonomous surface vehicle are setpoint regulation or path tracking tasks, the sensible choice for the flat output candidate would be the vector of position in the horizontal plane $\boldsymbol{z} = [x\ y]^\top$. Because the system has only two independent inputs, the flat output must have exactly two components. It is therefore impossible to consider the two position coordinates as well as the orientation

ψ as the flat output.

As stated in both [135] and [154], the general model of the surface vessel presented in equation (4.16) is not flat. Trying to express all systems variables as functions of $\boldsymbol{z}$ and its time derivatives, one quickly realizes that no algebraic expression of the ψ angle can be found. A differential expression is mandatory. Using the first time derivative of equation (4.16a) and equation (4.16b) and trigonometric combinations, the following relation is established:

$$\ddot{x}\sin\psi - \ddot{y}\cos\psi = u\dot{\psi} - \frac{1}{m - Y_{\dot{v}}}\left(-(m - X_{\dot{u}})u\dot{\psi} + Y_v v + Y_{\dot{r}}\ddot{\psi} + Y_r\dot{\psi}\right) \tag{4.17}$$

Equation (4.17) is a second order differential equation in yaw angle ψ. The differential equations preventing the flatness demonstration are referred to as *flatness defects* [53]. The defect of a non flat system is the number of differential equations obtained when trying to show flatness of a system. Here, the system as a defect equal to 1.

Remark 4.1 *Studying equation* (4.17) *also allows finding the* defects *preventing flatness of the system. As said before, flatness is an inherent property of a physical system but it can depend on numerical values of some constant parameters of the plant. For a given system as equation* (4.16)*, some sets of numerical parameters can cancel the defects of a system and allow showing flatness while another set will give a non-flat system* [1]*. Notably, looking at equation* (4.17)*, it appears clearly that the off-diagonal terms of the added mass and damping matrices $Y_{\dot{r}}$ and Y_r bring derivatives of the yaw angle in equation* (4.16d) *and equation* (4.16e)*. Choosing to neglect these terms or working with systems where they are naturally null would modify the flatness property of the system. This is the case of the Hovercraft system studied in the following.*

Note that [154] presents a simplified version of the SV model as a *Liouvillian* system which is one of the *almost* flat properties of a system that can be used to calculate controllers. Here, this more complete expression of the plant cannot be shown to be Liouvillian.

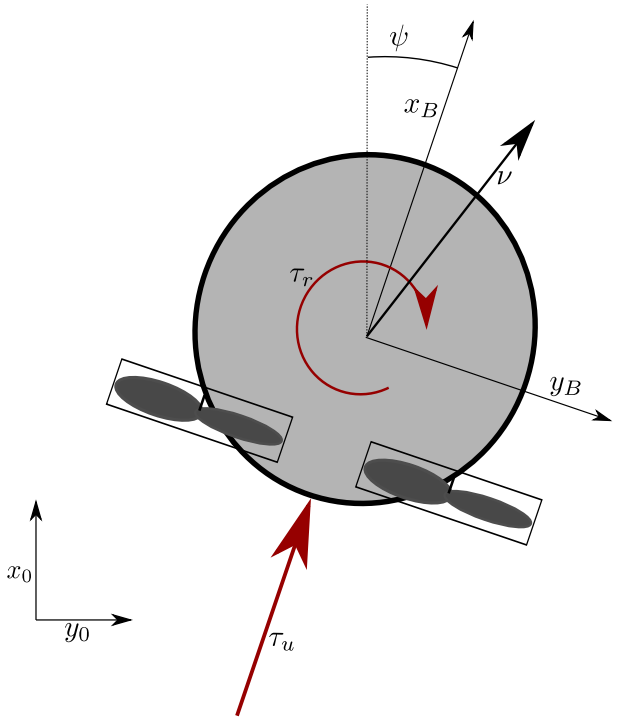


Figure 4.1 – Simplified representation of the hovercraft.

4.3.3 Special case: the Hovercraft system

The hovercraft system is a typical case study of mobile robots. It is described as an underactuated surface vehicle controlled in surge and yaw like the surface vessel studied in the previous section but with the distinctive feature of being of circular shape with homogeneous mass distribution. This property of the hovercraft system translates in the added mass and damping matrices. The coupling terms between sway and yaw are canceled in both matrices: $Y_{\dot{r}} = N_{\dot{v}} = 0$ and $Y_r = N_v = 0$. Also, the circular shape of the vehicle implies that the hydrodynamic effects of damping and added mass on the surge and sway axes are the same, giving $X_{\dot{u}} = Y_{\dot{v}}$ and [2] $X_u = Y_v$. The matrices of (4.15)

1. To our best knowledge, this fact only appears implicitly in previous works about differential flatness.
2. Some work like [154] use $X_u = 0$ and $Y_v \neq 0$ and obtain similar results.

become:

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - X_{\dot{u}} & 0 \\ 0 & 0 & I_0 - N_{\dot{r}} \end{bmatrix} \tag{4.18a}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -(m - X_{\dot{u}})v \\ 0 & 0 & (m - X_{\dot{u}})u \\ (m - X_{\dot{u}})v & -(m - X_{\dot{u}})u & 0 \end{bmatrix} \tag{4.18b}$$

$$\boldsymbol{D}(\boldsymbol{\nu}) = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{4.18c}$$

With these new hypotheses, the surface vessel system becomes:

$$\dot{x} = u\cos\psi - v\sin\psi \tag{4.19a}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{4.19b}$$

$$\dot{\psi} = r \tag{4.19c}$$

$$\dot{u} = \frac{\tau_u}{m - X_{\dot{u}}} + vr + \frac{X_u}{m - X_{\dot{u}}}u \tag{4.19d}$$

$$\dot{v} = -ur + \frac{Y_v}{m - Y_{\dot{v}}}v \tag{4.19e}$$

$$\dot{r} = \frac{\tau_r}{I_0 - N_{\dot{r}}} + \frac{N_r}{I_0 - N_{\dot{r}}}r \tag{4.19f}$$

Notice that many coupling terms have been canceled in the new expression (4.19) like in the model of the cylindrical AUV used before in this work (see chapter 1 for details). Notably, the yaw dynamics (4.19f) are now independent of the two other DOF and some yaw speed and acceleration terms have been canceled out from the surge and sway dynamics, equation (4.19d) and equation (4.19e) respectively.

For ease of writing, the following notations are introduced:

$$\beta = \frac{X_u}{m - X_{\dot{u}}} = \frac{Y_v}{m - Y_{\dot{v}}} \tag{4.20a}$$

$$\gamma = \frac{N_r}{I_0 - N_{\dot{r}}} \tag{4.20b}$$

$$\tilde{\tau}_u = \frac{\tau_u}{m - X_{\dot{u}}} \tag{4.20c}$$

$$\tilde{\tau}_r = \frac{\tau_r}{I_0 - N_{\dot{r}}} \tag{4.20d}$$

The model thus reads:

$$\dot{x} = u\cos\psi - v\sin\psi \tag{4.21a}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{4.21b}$$

$$\dot{\psi} = r \tag{4.21c}$$

$$\dot{u} = \tilde{\tau}_u + vr + \beta u \tag{4.21d}$$

$$\dot{v} = -ur + \beta v \tag{4.21e}$$

$$\dot{r} = \tilde{\tau}_r + \gamma r \tag{4.21f}$$

4.3.3.1 Flatness of the hovercraft

As before, the sensitive flat output candidate is $\boldsymbol{z} = [x\ y]^\top$ as it would enable straightforward position tracking applications. In the system (4.21) the variables which need to be expressed as functions of the flat output candidate to demonstrate flatness of the system are the yaw angle ψ, the surge and sway speeds u and v and the two system inputs $\tilde{\tau}_u$ and $\tilde{\tau}_r$.

Some of the interesting steps leading to the demonstration of flatness are presented here. First the kinematic part of the model is inverted to give expressions of the speed of the vehicle in the mobile frame:

$$u = \dot{x}\cos\psi + \dot{y}\sin\psi \tag{4.22a}$$

$$v = -\dot{x}\sin\psi + \dot{y}\cos\psi \tag{4.22b}$$

Then, the trigonometric combinations of the first time derivatives of equation (4.21a)

and equation (4.21b) are used to find formulations of ψ and $\tilde{\tau}_u$.

$$\ddot{x}\cos\psi + \ddot{y}\sin\psi = \tilde{\tau}_u + \beta u \tag{4.23a}$$
$$-\ddot{x}\sin\psi + \ddot{y}\cos\psi = \beta v \tag{4.23b}$$

Equation (4.23b) and (4.22b) directly give:

$$\tan\psi = \frac{\ddot{y} - \beta\dot{y}}{\ddot{x} - \beta\dot{x}} \Rightarrow \psi = \text{ atan}\left(\frac{\ddot{y} - \beta\dot{y}}{\ddot{x} - \beta\dot{x}}\right) = \text{ atan}\left(\frac{Y}{X}\right) \tag{4.24}$$

with $X = \ddot{x} - \beta\dot{x}$ and $Y = \ddot{y} - \beta\dot{y}$.

This first result is promising when compared to the general surface vehicle and equation (4.17). It shows that, because of the mass distribution and shape of the vehicle, the yaw angle is directly related to the velocity and acceleration of the vehicle in the inertial frame. This seems sensible considering the hypotheses, the vehicle naturally lines up with the vector of acceleration.

The expression of ψ (4.24) then gives:

$$\cos\psi = \frac{X}{\sqrt{X^2 + Y^2}} \tag{4.25a}$$
$$\sin\psi = \frac{Y}{\sqrt{X^2 + Y^2}} \tag{4.25b}$$

which in turn gives:

$$u = \frac{\dot{x}X + \dot{y}Y}{\sqrt{X^2 + Y^2}} \tag{4.26a}$$
$$v = \frac{-\dot{x}Y + \dot{y}X}{\sqrt{X^2 + Y^2}} \tag{4.26b}$$

Then, using equation (4.22a), equation (4.23a) and equation (4.25), the first input $\tilde{\tau}_u$ is calculated as:

$$\tilde{\tau}_u = \sqrt{X^2 + Y^2} \tag{4.27}$$

Finally, because the yaw speed and acceleration are decoupled from the two other

DOF of the system, successive differentiation of equation (4.24) gives:

$$r = \frac{\dot{Y}X - \dot{X}Y}{X^2 + Y^2} \tag{4.28a}$$

$$\tilde{\tau}_r = \frac{\ddot{Y}X - \ddot{X}Y}{X^2 + Y^2} - \frac{2(\dot{Y}X - \dot{X}Y)(X\dot{X} + Y\dot{Y})}{(X^2 + Y^2)^2} + \gamma\frac{\dot{Y}X - \dot{X}Y}{X^2 + Y^2} \tag{4.28b}$$

All the variables of the system have been expressed as functions of the flat output candidate $\boldsymbol{z} = [x \ y]^\top$ and its time derivatives up to the fourth order. The hovercraft model (4.21) is thus flat and $\boldsymbol{z}$ is indeed a flat output. At this point, the trajectory of the flat output must be continuously defined up to the fourth order. As seen later on, change of input allows reducing the need to the second order.

The main differences between the hovercraft model (4.21) and the general model of the underactuated surface vessel (4.16) are the hypotheses made on the numerical values of the model parameters. This goes to show that the flatness property of a system can depend on the numerical value of some parameters. Some sets of parameters can cancel or create defects preventing flatness.

4.3.3.2 Flatness-based controller of the hovercraft

One special feature of the hovercraft plant (4.21) is that the second input $\tilde{\tau}_r$ only controls yaw and is decoupled from the two other DOF of the system. Equation (4.21f) can be written as:

$$\tilde{\tau}_r = \ddot{\psi} - \gamma\dot{\psi} \tag{4.29}$$

This last expression can be thought as a change of control input: instead of considering the yaw moment as an input, the yaw angle will be used as input of the simplified model[3]:

$$\dot{x} = u\cos\psi - v\sin\psi \tag{4.30a}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{4.30b}$$

$$\dot{u} = \tilde{\tau}_u + v\dot{\psi} + \beta u \tag{4.30c}$$

$$\dot{v} = -u\dot{\psi} + \beta v \tag{4.30d}$$

3. This method is different from the two examples [154] and [135] both using second order state expansion on $\tilde{\tau}_u$ to create two fourth-order input equations of the flat output.

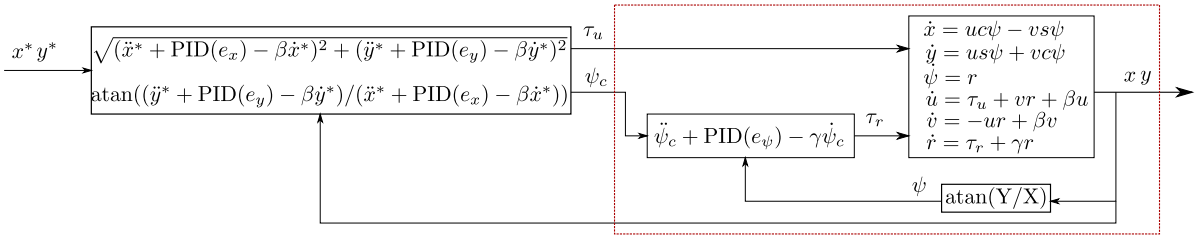


Figure 4.2 – Bloc diagram of the flatness-based controller

where the two inputs $\tilde{\tau}_u$ and ψ can be expressed as:

$$\tilde{\tau}_u = \sqrt{(\ddot{x} - \beta\dot{x})^2 + (\ddot{y} - \beta\dot{y})^2} \tag{4.31a}$$

$$\psi = \text{atan}\left(\frac{\ddot{y} - \beta\dot{y}}{\ddot{x} - \beta\dot{x}}\right) \tag{4.31b}$$

Remark 4.2 *System* (4.30) *does not have the standard form of non-linear systems: the first order derivative of one of the input components $\dot{\psi}$ appears in the last two equations. This system thus belong to the class of* generalized state *systems. The interested reader is referred to [38] and the references therein for more information about this class of systems. While the input derivative is naturally canceled in the following development, an alternative representation of system* (4.30) *is given in appendix B without any derived input [47].*

In this example, because no favored output component appear in equation (4.31), the controller is designed using the Brunovský state of the system [38, 84]. Following the example of section 4.1, the following state transform is a sensitive Brunovský state candidate:

$$\xi_{1,1} = x \tag{4.32a}$$

$$\xi_{1,2} = \dot{x} \tag{4.32b}$$

$$\xi_{2,1} = y \tag{4.32c}$$

$$\xi_{2,2} = \dot{y} \tag{4.32d}$$

$$\tag{4.32e}$$

Using state transform (4.32), system (4.30) becomes:

$$\dot{\xi}_{1,1} = \xi_{1,2} \tag{4.33a}$$
$$\dot{\xi}_{1,2} = \tilde{\tau}_u \cos\psi + \beta\xi_{1,2} \tag{4.33b}$$
$$\dot{\xi}_{2,1} = \xi_{2,2} \tag{4.33c}$$
$$\dot{\xi}_{2,2} = \tilde{\tau}_u \sin\psi + \beta\xi_{2,2} \tag{4.33d}$$

A new input $\boldsymbol{v} = [v_1\ v_2]^\top$ is introduced such as:

$$v_1 = \tilde{\tau}_u \cos\psi \tag{4.34a}$$
$$v_2 = \tilde{\tau}_u \sin\psi \tag{4.34b}$$

which yields a single solution for the original inputs:

$$\psi_c = \operatorname{atan}\left(\frac{v_2}{v_1}\right) \tag{4.35a}$$
$$\tilde{\tau}_{u,c} = v_1 \cos\psi_c + v_2 \sin\psi_c \tag{4.35b}$$

Equation (4.33) is then rewritten with the new input:

$$\dot{\xi}_{1,1} = \xi_{1,2} \tag{4.36a}$$
$$\dot{\xi}_{1,2} = v_1 + \beta\xi_{1,2} \tag{4.36b}$$
$$\dot{\xi}_{2,1} = \xi_{2,2} \tag{4.36c}$$
$$\dot{\xi}_{2,2} = v_2 + \beta\xi_{2,2} \tag{4.36d}$$

System (4.36) is expressed in the *controller form* [38]. An obvious simple feedback leads to the well known Brunovský form composed of two pure integrator chains, the details are left to the reader.

Note that, in the general case, the Brunovský state transform may depend on the input and, in the case of generalized state systems, on the input derivatives. It is a specificity of this system to be generalized in its original representation equation (4.30) and standard in its Brunovský form; All derivatives of the input disappear during the state transformation.

In an exact feedforward linearizing fashion [84], the controller is calculated on this

system as:

$$v_1 = \dot{\xi}^*_{1,2} + \text{PID}(e_1) - \beta\xi^*_{1,2} \tag{4.37a}$$
$$v_2 = \dot{\xi}^*_{2,2} + \text{PID}(e_2) - \beta\xi^*_{2,2} \tag{4.37b}$$

where PID is a standard PID controller function and $e_1 = \xi^*_{1,1} - \xi_{1,1}$ and $e_2 = \xi^*_{2,1} - \xi_{2,1}$ are the tracking errors. As before the upperscript $*$ marks desired values of the variable. The controller (4.37) gives the closed-loop system:

$$\ddot{e}_1 + k_D\dot{e}_1 + k_P e_1 + k_I\chi_1 - \beta\dot{e}_1 = 0 \tag{4.38a}$$
$$\ddot{e}_2 + k_D\dot{e}_2 + k_P e_2 + k_I\chi_2 - \beta\dot{e}_2 = 0 \tag{4.38b}$$

where $\chi_i = \int_0^t e_i(\sigma)d\sigma$ and k_P, k_D and k_I are the positive gain parameters o the PID. In the present case, the closed-loop system is linear. The feedforward terms of the controller add the two damping terms $\beta\dot{e}_1$ and $\beta\dot{e}_2$ in the error dynamics. The damping parameter $\beta < 0$ thus behaves as an additional derivative gain stabilizing the system. In this example, the same gains have been chosen on the two axes as the vehicle is perfectly symmetrical. In the following applications, notably when the hypothesis of a circular hull is not met, different sets of gains on the two axes could be used.

Here, the original inputs of system (4.30) can be calculated using equation (4.35):

$$\begin{aligned}\psi_c &= \operatorname{atan}\left(\frac{\dot{\xi}^*_{2,2} + \text{PID}(e_2) - \beta\xi^*_{2,2}}{\dot{\xi}^*_{1,2} + \text{PID}(e_1) - \beta\xi^*_{1,2}}\right) \\ &= \operatorname{atan}\left(\frac{\ddot{y}^* + \text{PID}(e_y) - \beta\dot{y}^*}{\ddot{x}^* + \text{PID}(e_x) - \beta\dot{x}^*}\right)\end{aligned} \tag{4.39a}$$

$$\begin{aligned}\tilde{\tau}_{u,c} &= (\dot{\xi}^*_{1,2} + \text{PID}(e_1) - \beta\xi^*_{1,2})\cos\psi_c + (\dot{\xi}^*_{2,2} + \text{PID}(e_2) - \beta\xi^*_{2,2})\sin\psi_c \\ &= (\ddot{x}^* + \text{PID}(e_x) - \beta\dot{x}^*)\cos\psi_c + (\ddot{y}^* + \text{PID}(e_y) - \beta\dot{y}^*)\sin\psi_c\end{aligned} \tag{4.39b}$$

The final controller (4.39) indeed shows the same symmetry characteristics as the rest of the hovercraft system. Because no input has been favored in the inputs calculations, the controllability of the system is ensured.

Remark 4.3 *Note that the final controller* (4.39) *has the same shape as the flatness input equations* (4.31)*. While it would have not been rigorous to directly use the equations of flatness to build the controller on this system for the various reasons exposed before, the*

alternative method without using the Brunovský state would have given a similar controller structure.

Remark 4.4 *In the presence of yaw disturbance or when considering actuators limitations, it is necessary to add an extra level of control on yaw. The yaw subsystem (Eqs.* (4.21c), (4.21f)*) being flat with ψ as the flat output, the yaw controller can be:*

$$\tilde{\tau}_{r,c} = \dot{r}^* + k_d^\psi \dot{e}_\psi + k_p^\psi e_\psi + k_i^\psi \chi_\psi - \gamma r^* \tag{4.40}$$

with $e_\psi = \psi_c - \psi$, $r^* = \dot{\psi}_c$, $\dot{r}^* = \ddot{\psi}_c$ and k_p^ψ, k_i^ψ and k_d^ψ are appropriate gain parameters. The yaw angle ψ can be measured or calculated using the equation of flatness (4.24)*. Any other autopilot could be used here provided that they are faster than the main flatness-based control loop to ensure the behavior of the cascaded system.*

Before adding the iPID in the flatness-based controller, it is tested on a SV model with a conventional PID. The work of [85] shows that flatness-based control is naturally robust to a certain amount of model approximation (10% in the reference) and that the approximation of some of the terms of the model can act favorably for the stability of the closed-loop system. In is then reasonable to expect the controller (4.39) to give good result when applied on the SV. Of course in the SV model, added-mass and damping are different on the surge and sway axis: $X_{\dot{u}} \neq Y_{\dot{v}}$ and $X_u \neq Y_v$. The parameter β of equation (4.39) would then be chosen as the smaller of the two damping parameters so that the controller does not rely on a false natural damping effect. The simulation results presented in section 4.5.2 show that the flatness based controller (4.39) associated with the inner loop (4.40) gives very good trajectory tracking results on both the hovercraft and the surface vessel. The main limits of this controller as demonstrated in section 4.5.2.4 is the behavior outside of the neighborhood of the the trajectory. This weakness is compensated in the following section with an additional layer of Intelligent PID control.

4.4 Association with Intelligent PID

In this Section, the results obtained on the hovercraft system are extended to all surface vessels. To do so, the flatness-based controller developed in the previous section is associated with Model Free Control and Intelligent PIDs (iPIDs). The idea is to ally the natural robustness of flatness-based control with the adaptability of the iPID to design

a controller which can be applied to any surface vessel. Therefore, the controller is calculated as if the system to control were a hovercaft (with the hypotheses of circular shape and symmetrical mass distribution) but is applied to any surface vessel model.

As exposed in section 2.5.1, iPIDs are a wide-spread model-free control solution successfully applied to a multitude of various systems. Details about the theory behind iPIDs can be found in [49]. The idea is to approximate the system with an ultre-local model containing an adaptive term estimating all the non-modeled effects acting on the system as a single number (or vector). Here, the iPID is associated with the nominal control calculated with flatness, is applied to the Brunovský state representation of the hovercraft system and then propagated in the model of the vehicle.

Intelligent PIDs have yet to be implemented on the underactuated SV. The SV being ill-actuated with respect to the task (lack of one translation), iPID alone will not be sufficient to complete the task. A compensation mechanism is needed to control the lacking translation using the rotation. It is therefore associated with flatness to create this compensation mechanism.

4.4.1 Implementation of the iPID

Recall the Brunovský representation of the hovercraft:

$$\dot{\xi}_{1,1} = \xi_{1,2} \tag{4.41a}$$
$$\dot{\xi}_{1,2} = v_1 + \beta\xi_{1,2} \tag{4.41b}$$
$$\dot{\xi}_{2,1} = \xi_{2,2} \tag{4.41c}$$
$$\dot{\xi}_{2,2} = v_2 + \beta\xi_{2,2} \tag{4.41d}$$

with $\xi_{1,1} = x$, $\xi_{1,2} = \dot{x}$, $\xi_{2,1} = y$ and $\xi_{2,2} = \dot{y}$.

In order to associate the flatness-based controller calculated in the previous section with the iPID, the control inputs are broken down into two parts:

$$v_1 = v_1^* + \delta v_1 \tag{4.42a}$$
$$v_2 = v_2^* + \delta v_2 \tag{4.42b}$$

where v_1^* and v_2^* are the nominal controller calculated with flatness and δv_1 and δv_2 are the iPID contribution. The idea here is to use the iPID to control the vehicle around

its nominal trajectories dictated by the flatness-based controller. This is somewhat similar to the idea behind SMC where the system is oscillating around the sliding surface representing the desired equilibrium.

Considering the system (4.41), the nominal flatness-based controllers are given as:

$$v_1^* = \dot{\xi}_{1,2}^* - \beta \xi_{1,2}^* = \ddot{x}^* - \beta \dot{x}^* \tag{4.43a}$$
$$v_2^* = \dot{\xi}_{2,2}^* - \beta \xi_{2,2}^* = \ddot{y}^* - \beta \dot{y}^* \tag{4.43b}$$
$$\tag{4.43c}$$

Then, applying the controller (4.42) into the system (4.41), a new system based on the tracking error and controlled with the iPID controllers δv_1, δv_2 is created:

$$\ddot{x} = v_1^* + \delta v_1 + \beta \dot{x} = \ddot{x}^* - \beta \dot{x}^* + \delta v_1 + \beta \dot{x} \Rightarrow \ddot{e}_x = \beta \dot{e}_x - \delta v_1 \tag{4.44a}$$
$$\ddot{y} = v_2^* + \delta v_2 + \beta \dot{y} = \ddot{y}^* - \beta \dot{y}^* + \delta v_2 + \beta \dot{y} \Rightarrow \ddot{e}_y = \beta \dot{e}_y - \delta v_2 \tag{4.44b}$$

where the subscript $*$ denotes desired values of the variable and $e_x = x^* - x$ and $e_y = y^* - y$.

Two iPID controllers are therefore designed to control the error system in the vicinity of the trajectory. The system is composed of two second order decoupled subsystems thus second order ultra-local models are used. The two error signals are considered as measurements in the two ultra-local models:

$$\ddot{e}_x = \alpha_x \delta v_1 + F_x \tag{4.45a}$$
$$\ddot{e}_y = \alpha_y \delta v_2 + F_y \tag{4.45b}$$

where F_x and F_y are the two adaptive functions estimating the unmodeled terms and α_x and α_y are two control parameters.

Following the reasoning in section 2.5, the estimation of F_x and F_y is done using ALIEN filters [58, 57]. For the second order ultra-local models, F_x and F_y are estimated as:

$$\widehat{F}_x = \frac{5!}{2T^5} \int_0^T \left[\left[(T-\tau)^2 - 4(T-\tau)\tau + \tau^2 \right] x(\tau) - \alpha_x \frac{(T-\tau)^2}{2} \tau^2 \delta v_1(\tau) \right] d\tau \tag{4.46a}$$
$$\widehat{F}_y = \frac{5!}{2T^5} \int_0^T \left[\left[(T-\tau)^2 - 4(T-\tau)\tau + \tau^2 \right] y(\tau) - \alpha_y \frac{(T-\tau)^2}{2} \tau^2 \delta v_2(\tau) \right] d\tau \tag{4.46b}$$

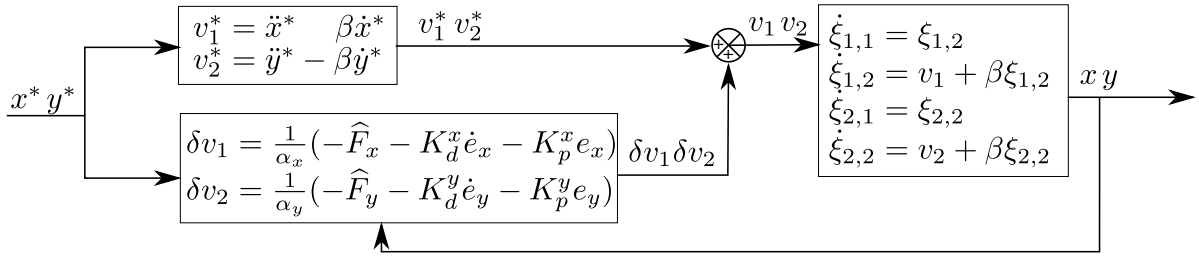


Figure 4.3 – Bloc diagram of the flatness and iPD controller applied to the Brunovskỳ state representation.

The adaptive terms $\widehat{F}_x$ and $\widehat{F}_y$ are considered constant on one time sample and are calculated once for each sample. The time dependency in in the expressions of $\widehat{F}_x$ and $\widehat{F}_y$ is omitted for clarity.

Because the two iPIDs are calculated on the error system, the desired values in both e_x and e_y are all zero: $e_x^* = \dot{e}_x^* = \ddot{e}_x^* = 0$ and $e_y^* = \dot{e}_y^* = \ddot{e}_y^* = 0$. The two control inputs of the ultra-local models are then:

$$\begin{aligned}\delta v_1 &= \frac{1}{\alpha_x}\left(\ddot{e}_x^* - \widehat{F}_x + K_d^x(\dot{e}_x^* - \dot{e_x}) + K_p^x(e_x^* - e_x)\right)\\ &= \frac{1}{\alpha_x}\left(-\widehat{F}_x + K_d^x(-\dot{e_x}) + K_p^x(-e_x)\right) \qquad (4.47a)\\ \delta v_2 &= \frac{1}{\alpha_y}\left(\ddot{e}_y^* - \widehat{F}_y + K_d^y(\dot{e}_y^* - \dot{e_y}) + K_p^y(e_y^* - e_y)\right)\\ &= \frac{1}{\alpha_y}\left(-\widehat{F}_y + K_d^y(-\dot{e_y}) + K_p^y(-e_y)\right) \qquad (4.47b)\end{aligned}$$

with K_D^x, K_P^x, K_D^y and K_P^y strictly positive gain parameters. The complete controller is thus:

$$v_1 = \ddot{x}^* - \beta\dot{x}^* + \frac{1}{\alpha_x}\left(-\widehat{F}_x + K_d^x(-\dot{e_x}) + K_p^x(-e_x)\right) \qquad (4.48a)$$

$$v_2 = \ddot{y}^* - \beta\dot{y}^* + \frac{1}{\alpha_y}\left(-\widehat{F}_y + K_d^y(-\dot{e_y}) + K_p^y(-e_y)\right) \qquad (4.48b)$$

Figure 4.3 shows the final flatness and iPD controller applied to the Brunovskỳ state representation of the system.

Note that different sets of gain parameters are used on the two axes. Without the even mass distribution approximation of the hovercraft system, the gain parameters have no reason to be the same on the two axes. Practically, they could be chosen equal

in some cases.

Also, no integral term is necessary in equation (4.48). The work of [127] and [48] shows that the adaptive terms F_x and F_y cover the work of the integral term of a conventional PID. In facts, intelligent proportional derivative controllers as this one are shown to be equivalent to conventional PIDs.

4.5 Simulations of the flatness-based controllers

This sections presents simulations of the two flatness-based controllers developed on the fully-actuated and underactuated vehicles. In the fully-actuated case, the flatness-based controller is compared with a feedback linearizing PID-based controller and a sliding mode controller. It shows very good trajectory tracking performance and robustness external disturbance.

In the underactuated case, the conventional flatness-based controller and the association with iPID are shown to be very effective on both the hovercraft and the generic surface vessel. Both controller show good robustness to external disturbances, measurement noise and model approximations. The iPID is shown to compensate the behavior of conventional flatness when the systems gets too far from the desired path.

4.5.1 Fully-actuated AUV

In the fully-actuated case, the flatness-based controller is compared with a feedback linearizing PID-based controller and a sliding mode controller. These controllers are compared on seabed scanning mission consisting in horizontal rails at constant depth. The task requires the six DOF, the center of the vehicle must actively track the position of the target moving on the trails as well as following the attitude trajectory. To confirm the theoretical performances of the flatness-based controller it is first tested on a reference scenario with no external disturbance or approximation and then evaluated with external disturbances, estimation errors of the model parameters and initial errors.

The control laws used for comparison are recalled here. The first controller is a standard feedback linearizing, PID-based controller calculated in the inertial frame.

$$\boldsymbol{\tau}_{PID} = \boldsymbol{M}\left(\boldsymbol{J}(\boldsymbol{\eta})^{-1}(\ddot{\boldsymbol{\eta}}^{*} + \mathbf{PID}(\boldsymbol{e}_{\eta})) + \dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1}\dot{\boldsymbol{\eta}}\right) + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \quad (4.49)$$

In equation (4.49), $\mathbf{PID}(\boldsymbol{e}_\eta)$ denotes six decoupled PID controllers, one per DOF, calculated in the inertial frame on the position and orientation errors contained in $\boldsymbol{e}_\eta$. In this example, the same gain parameters are used on all six DOF. Allegedly, using a different set of gain parameters on each DOF would allow finer tuning of the performances on each axis. Yet, the same of gains used in this example shows very good performances on all six DOF.

The second control law is the Sliding Mode controller presented in section 2.4 [40] and adapted to this vehicle. It is composed of six decoupled sliding mode controllers, one per DOF. To avoid chattering and as is usually done, the signum function was replaced by a saturation function. Note that, to match with the greater number of references, the control surfaces are calculated with respect to the position and orientation error vector. No integral term is added and therefore steady state error can appear in the presence of unknown external disturbances (See appendix A).

$$\boldsymbol{\tau}_{SMC} = \tilde{\boldsymbol{M}}(\boldsymbol{\eta})\left(\ddot{\boldsymbol{\eta}}^* + \boldsymbol{\lambda}\dot{\boldsymbol{e}}_\eta + \boldsymbol{K}\,\mathrm{sat}(\boldsymbol{\sigma})\right) + \tilde{\boldsymbol{C}}(\boldsymbol{\eta},\boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{D}}(\boldsymbol{\eta},\boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{4.50}$$

In this example, the gain matrices $\boldsymbol{\lambda}$ and $\boldsymbol{K}$ are diagonal and the same gain parameters are used for all six DOF. The expressions of the model matrices expressed in the inertial frame $\tilde{\boldsymbol{M}}$, $\tilde{\boldsymbol{C}}$ and $\tilde{\boldsymbol{D}}$ are given in Equation (4.10).

For the sake of comparison, the flatness-based controller is recalled here:

$$\boldsymbol{\tau}_{Flat} = \tilde{\boldsymbol{M}}(\boldsymbol{\eta}^*)(\ddot{\boldsymbol{\eta}}^* + \mathbf{PID}(\boldsymbol{e}_\eta)) + \tilde{\boldsymbol{C}}(\boldsymbol{\eta}^*,\boldsymbol{\nu}^*)\dot{\boldsymbol{\eta}}^* + \tilde{\boldsymbol{D}}(\boldsymbol{\eta}^*,\boldsymbol{\nu}^*)\dot{\boldsymbol{\eta}}^* + \boldsymbol{g}(\boldsymbol{\eta}^*) \tag{4.51}$$

The first difference appearing between the three controllers is the linearizing terms. As described before, one of the main assets of flatness-based control is that the conventional linearizing terms of a feedback linearizing controller are substituted with the nominal value of the controller, creating increased robustness. One the other hand, the linearizing terms of both the PID-based controller and the SM controller are calculated at the current state of the vehicle. Besides, all three controllers show very similar structures. The three control laws are calculated using error vectors expressed in the inertial frame.

All the following tests are performed on the *RSM* vehicle equipped with six independent fixed thrusters and presented in figure 1.3. The control parameters of the three controllers are given in table 4.1. All the gain matrices are diagonal with equal factors on all the rows.

Item	**Value**	**Item**	**Value**
Trajectory	Seabed Scanning	**Nominal Speed**	$1.5m.s^{-1}$
Simulation time sample	$0.05ms$	**Vehicle Shape**	Cylinder
Flatness-based controller			
k_p	20	k_i	5
k_d	8		
PID			
k_p	20	k_i	5
k_d	8		
Sliding Mode Controller			
K	5	λ	10

Table 4.1 – Simulation parameters - fully actuated vehicle

The gain parameters of the PID-based controller and those of he Flatness-based controllers have been purposely chosen equal to allow better comparison of the linearizing techniques. The nominal speed of the following tests has been chosen to ensure that the thrusters of these vehicles stay far from their maximum thrusts. At nominal speed, the thrusters generate about 35% of their maximal thrust.

It is worth mentioning that all three controllers have equivalent energy consumption on each of the following tasks.

4.5.1.1 Base case scenario

The base case scenario of this study is a trajectory tracking task with no external disturbance or model approximation. The vehicle is placed on the task at the initial time so there is no initial error. In facts, the trajectory is composed of six independent trajectories, one per DOF, continuously defined up to the fourth order. They are described with connected polynomials to ensure continuity up to the fourth order.

Figures 4.4 and 4.5 respectively show the position and orientation errors of the three controllers. As expected, the three controllers perform equally well on the six DOF during the complete experiment. Very slightly larger errors appear on the $\boldsymbol{x}_{\mathbf{o}}$ and

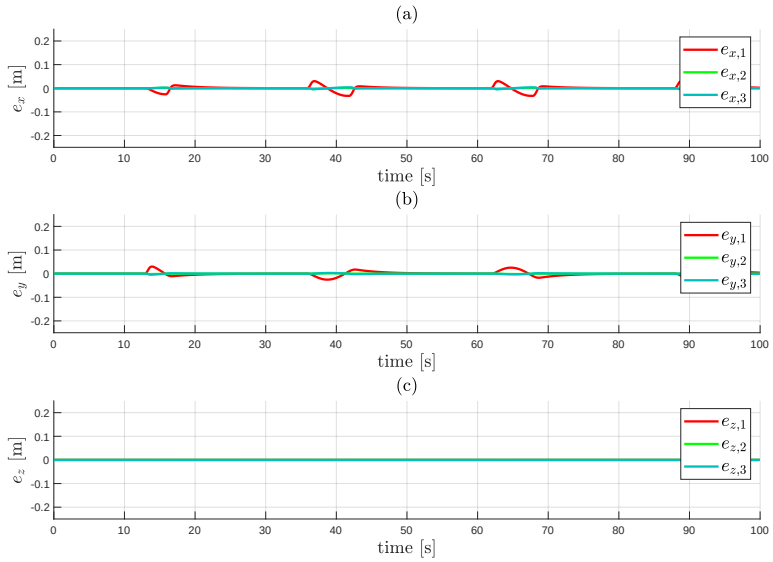


Figure 4.4 – Comparison of the position errors for the three controllers. Red: PID - Green: SMC - Blue: Flatness-based

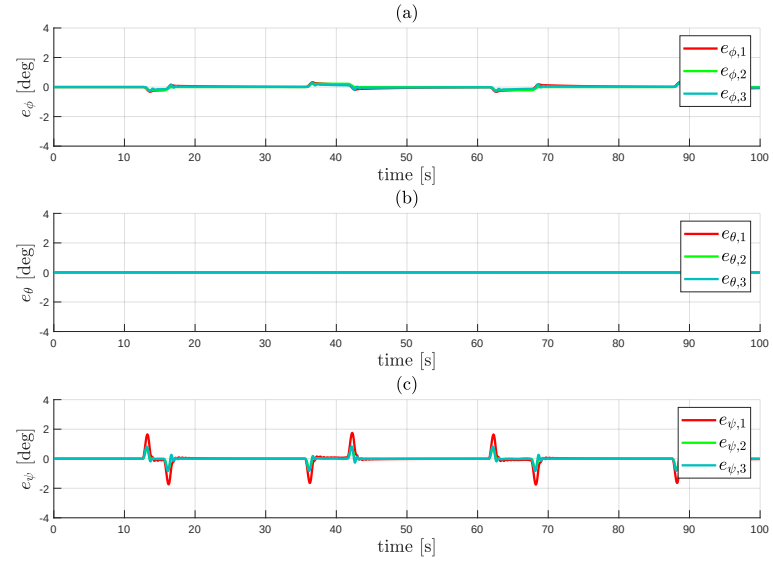


Figure 4.5 – Comparison of the orientation errors for the three controllers. Red: PID - Green: SMC - Blue: Flatness-based

$\boldsymbol{y_o}$ axis of the PID-based controller (Red line on figure 4.4.(a) and figure 4.4.(b)) and on the yaw error (Red line on figure 4.5.(c)). Considering that the gain parameters are the same for the FBLN and Flatness-based controllers, these difference (magnitudes of a centimeter and a degree respectively) may be attributed to the linearizing terms.

4.5.1.2 Initial error

As exposed in the previous sections, Flatness-based control has notably been designed to be used at the vicinity of the desired trajectory. Using trajectory values in the linearizing terms of the controller implies that the nonlinearities of the closed-loop system cancel only when the system is on trajectory. Otherwise, some non linear terms remain in the closed-loop system.

To illustrate the behavior of the Flatness-based controller when the system is not on the trajectory, two tests are performed with initial position error on the $\boldsymbol{y_o}$ axis. For the first test, depicted on figure 4.6, the initial error is $1\,\mathrm{m}$ while for the second test depicted on figure 4.7, the initial error is $5\,\mathrm{m}$. In the initial configuration, the error is on the $\boldsymbol{y_B}$ axis of the mobile frame, all other DOF (up to the fourth order) are initiated at the desired value. The vehicle is then supposed to perform a sway motion whilst maintaining proper depth, surge speed and orientation.

In the first case, with the 1m initial error, all three vehicle converge on the trajectory. It is worth noting that the Flatness-based and PID-based controllers behave almost equally and converge faster than the SM controller but at the cost of a greater disturbance on the $\boldsymbol{x_o}$ axis (and greater orientation disturbances, not given here).

On the other hand, with the 5m initial error, the flatness-based controller does not converge on the trajectory. In facts, the $\boldsymbol{y_o}$ is not canceled but oscillates around the initial value, slightly diverging. More concerning, the controller generates great disturbances on other DOF as shown on figure 4.7.(a) and figure 4.7.(c): both errors on the $\boldsymbol{x_o}$ and $\boldsymbol{z_o}$ axes diverge. The PID-based and SM controllers eventually converge to the trajectory with the 5m initial error.

This example shows that the flatness-based controller performs as well as conventional controllers in the neighborhood of the trajectory but does not converge when the vehicle is too far from the trajectory. Of course, the numerical size of the boundary layer around the trajectory allowing convergence of the system depends on the system itself. Therefore, if the vehicle were to get outside of this convergence boundary around the trajectory (here, outside of a 5m range from the target), the flatness-based controller

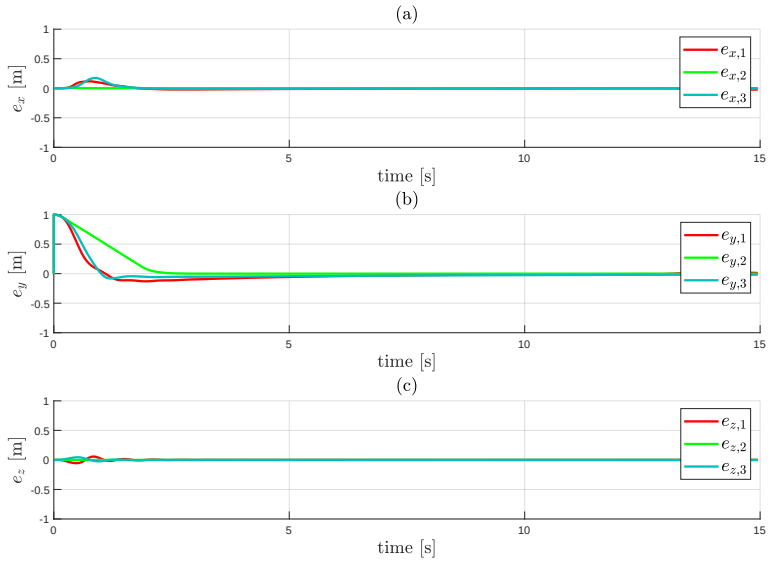


Figure 4.6 – Comparison of the position errors for the three controllers with a 1m initial error on the $\boldsymbol{y_o}$ axis. Red: PID - Green: SMC - Blue: Flatness-based

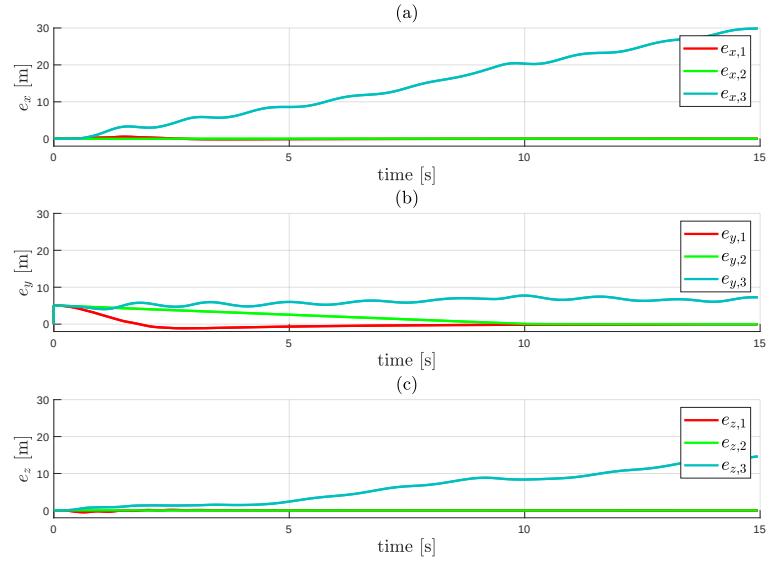


Figure 4.7 – Comparison of the orientation errors for the three controllers with a 5m initial error on the $\boldsymbol{y_o}$ axis. Red: PID - Green: SMC - Blue: Flatness-based

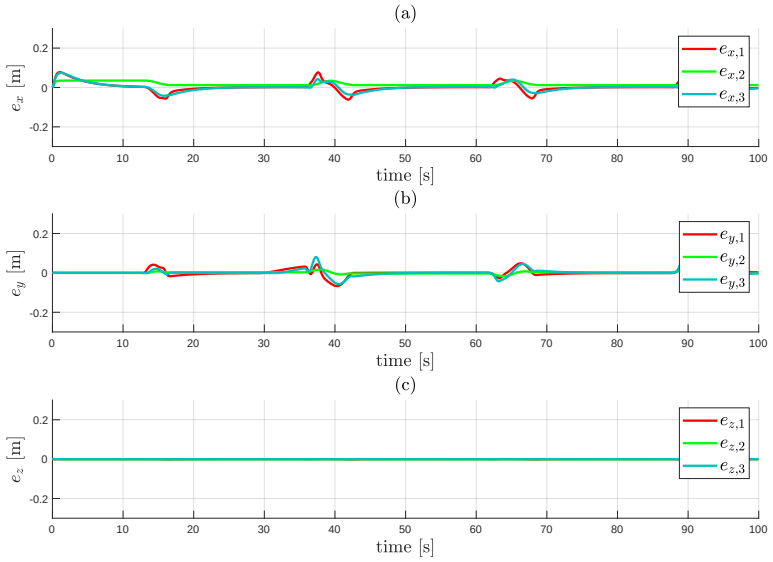


Figure 4.8 – Comparison of the position errors for the three controllers with a 0.75m/s current on $\boldsymbol{x_o}$ axis. Red: PID - Green: SMC - Blue: Flatness-based

would not allow correcting the error and a new trajectory should be calculated to catch up.

4.5.1.3 Robustness to external disturbances

To evaluate the robustness to external disturbances, an ocean current is added in simulation. This current is not known by the controller and therefore not compensated. In this example, a constant irrotational current is used. It is aligned with the $\boldsymbol{x_o}$ axis and its velocity is half of the nominal velocity required on the trajectory: 0.75 m/s.

As shown on figure 4.8 the PID-based controller and the flatness-based controller perform as well in this case. The errors shown on figure 4.8 for these two controllers are the same as in the base case figure 4.4. Note that the SM controller shows the expected steady state error on the $\boldsymbol{x_o}$ axis(Green line on figure 4.8.(a)).

Figure 4.9 shows that the flatness-based controller does not deviate much in roll (figure 4.9.(a)) but a 15 degrees deviation appears on the yaw angle on figure 4.9.(c). The yaw deviation is slightly larger than the one appearing with the PID-based controller. The PID-based controller also demonstrates some pitch oscillations when the vehicle is

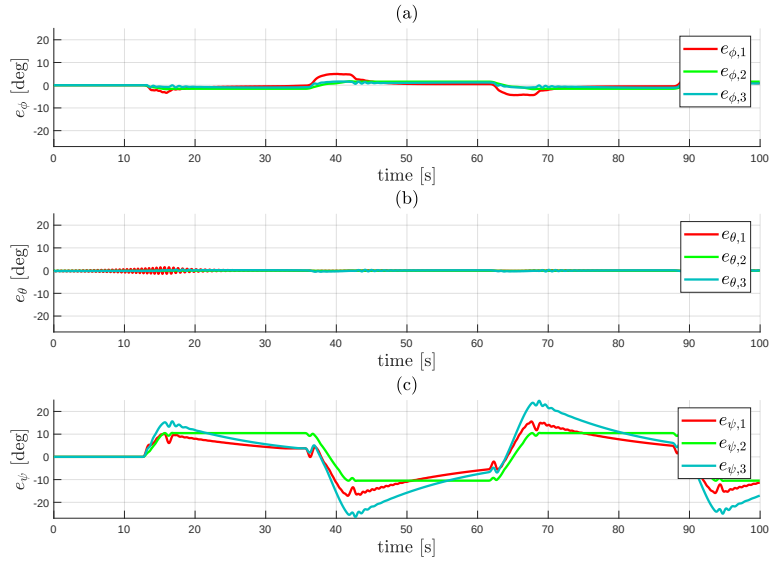


Figure 4.9 – Comparison of the orientation errors for the three controllers with a 0.75m/s current on $\boldsymbol{x_o}$ axis. Red: PID - Green: SMC - Blue: Flatness-based

facing the current that do not appear when using the Flatness-based controller. Here again, the SM controller shows steady state errors in both roll and yaw.

Overall, the Flatness-based controller shows good robustness to external disturbances in the fully-actuated case. In facts, as long as the vehicle stays in the vicinity of the trajectory, the integral term of the controller ensures convergence of the (almost linear) closed-loop system when facing unknown constant disturbances. The difference of pitch oscillations between the PID-based controller and the Flatness-based controller could be interpreted as a proof of the effect of the stabilizing non-linear terms resulting from the feedforward linearizing method. Further test are necessarily to evaluate the role of each of the terms of the closed-loop system created by the flatness-based controller but the simulations show that the flatness-based controller is as good as the other controllers from the literature.

4.5.2 Underactuated surface vessel

This section presents the trajectory tracking simulation results obtained with the two flatness-based controllers introduced in section 4.3.3 and section 4.4. Both controllers

are calculated using the hovercraft model considering a circular hull shape and an even mass distribution. They are applied first to a hovercraft and then to the model of a generic surface vessel. The surface vessel does not meet the hypothesis of circular shape and mass distribution of the hovercraft. The conventional hypothesis for estimation of the added mass and damping parameters are used (see chapter 1 for details). One notable difference is that the surge and sway added masses and damping parameters are not equal on the surface vessel and additional coupling terms appear between the sway and yaw in both the mass and damping matrices. The obvious flatness defects due to the shape of the surface vessel have been described in section 4.3.

The control laws are recalled here. Both controllers are calculated on the Brunovský representation of the system and then transformed into the inputs of the original system. The conventional flatness-based controller is:

$$v_1 = \ddot{x}^* + \text{PID}(e_x) - \beta\dot{x}^* \tag{4.52a}$$

$$v_2 = \ddot{y}^* + \text{PID}(e_y) - \beta\dot{y}^* \tag{4.52b}$$

where PID is a regular PID function with strictly positive gain parameters.

The second controller is the association of a flatness-based controller and an iPID. The inputs are calculated as the sum of the nominal control based on flatness and the iPID applied to the error system. The controller reads:

$$v_1 = \ddot{x}^* - \beta\dot{x}^* + \frac{1}{\alpha_x}\left(-\widehat{F}_x + K_d^x(-\dot{e_x}) + K_p^x(-e_x)\right) \tag{4.53a}$$

$$v_2 = \ddot{y}^* - \beta\dot{y}^* + \frac{1}{\alpha_y}\left(-\widehat{F}_y + K_d^y(-\dot{e_y}) + K_p^y(-e_y)\right) \tag{4.53b}$$

$$\widehat{F}_x = \frac{5!}{2T^5}\int_0^T \left[\left[(T-\tau)^2 - 4(T-\tau)\tau + \tau^2\right] x(\tau) - \alpha_x \frac{(T-\tau)^2}{2}\tau^2\delta v_1(\tau)\right] d\tau \tag{4.53c}$$

$$\widehat{F}_y = \frac{5!}{2T^5}\int_0^T \left[\left[(T-\tau)^2 - 4(T-\tau)\tau + \tau^2\right] y(\tau) - \alpha_y \frac{(T-\tau)^2}{2}\tau^2\delta v_2(\tau)\right] d\tau \tag{4.53d}$$

Little literature exist on the tuning of the control parameters α_x and α_y. The rule of thumb is to choose α_x ad α_y so that the inputs δv_1 and δv_2 are approximately the same order of magnitude as second order derivative of the measurements $\ddot{e}_x$ and $\ddot{e}_y$ respectively. In this case, because the two iPIDs are calculated on error systems, the expected values for $\ddot{e}_x$ and $\ddot{e}_y$ is zero. The control parameters α_x and α_y have then been chosen equal to 1. The damping parameters are chosen as $\beta = -12$ and $\gamma = -4$.

The controls of the original system are then obtained using:

$$\psi_c = \operatorname{atan}\left(\frac{v_2}{v_1}\right) \tag{4.54a}$$

$$\tilde{\tau}_{u,c} = v_1 \cos\psi_c + v_2 \sin\psi_c \tag{4.54b}$$

$$\tilde{\tau}_{r,c} = \dot{r}^* + k_d^\psi \dot{e}_\psi + k_p^\psi e_\psi + k_i^\psi \chi_\psi - \gamma r^* \tag{4.54c}$$

Note that the following simulation results have been obtained on a much simpler simulator than the other simulations presented in this work. They are preliminary results of the methods but some more refined work is still needed. The simplified simulator does not take limitations of variation of inputs into account but the control efforts are upper bounded with the arbitrary chosen maximal values of $\tilde{\tau}_{u,c}^{max} = 100\,\mathrm{N}$ and $\tilde{\tau}_{r,c}^{max} = 100\,\mathrm{Nm}$.

4.5.2.1 Ideal case

Item	**Value**	**Item**	**Value**
Trajectory	Lawn Mower	**Nominal Speed**	$1m.s^{-1}$
Simulation time sample	$0.05ms$	**Vehicle Shape**	Circular
Conventional Flatness-based controller			
k_p	11.1	k_i	3.3
k_d	6.6	k_p^ψ	100
k_i^ψ	10	k_d^ψ	20
iPID			
α_x	1	α_y	1
K_P^x	11.1	K_p^y	11.1
K_d^x	6.6	K_D^y	6.6
k_p^ψ	100	k_i^ψ	10
k_d^ψ	20		

Table 4.2 – Simulation parameters - Ideal case Hovercraft

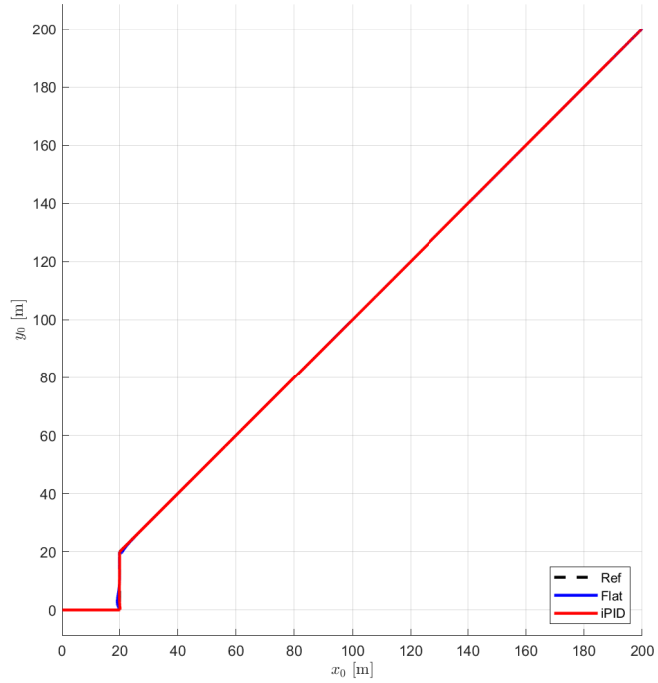


Figure 4.10 – Comparison of the trajectory of the flatness-based controllers in the ideal case - Red: conventional flatness, Blue: iPID, Black: Reference

The two controllers are first tested in the ideal case. They are applied to the hovercraft without noise and without external disturbance. The task is defined as a succession of three segments in the horizontal plane. The first one is aligned with the $\boldsymbol{x_o}$ axis, the second one with the $\boldsymbol{y_o}$ axis and the third one is diagonal. On each segment, the vehicle must accelerate to reach the nominal speed of $1\,\mathrm{m\,s^{-1}}$ then decelerate so that it arrives at the next turning point with zero velocity. This task is referred to as the *Lawn Mower* task referring to the behavior of an autonomous lawn mower robot.

The PID of the flatness-based controller, iPD and PID of the low-level ψ controller are tuned using a simplified pole placing method. The poles of each system are chosen to be in the sam place so that the three controller can be tuned choosing only one characteristic time parameter. The sets of gains are chosen to ensure that the inner loop controlling the ψ angle is always faster than the outer loop.

Figure 4.10 shows the behavior of the two vehicles on the task. The trajectory may

appear discontinuous on figure 4.10 but it is in fact continuous up to the fourth order. Both controllers allow almost perfect tracking of the trajectory in the ideal case as expected. Without disturbances, and applying the controllers to the hovercraft system, the adaptive functions F_x and F_y of the iPD have noting to estimate and converge to zero very quickly. Once they reach zero, both controllers are perfectly equivalent.

This example shows that the flatness-based structure of the controllers acts as a guidance method on the underactuated hovercraft. The calculation of the command angle ψ_c creates the compensation mechanism of underactuation mentioned in previous examples this time relying on the natural flatness of the plant.

4.5.2.2 External disturbance and Measurement noise

Item	**Value**	**Item**	**Value**
Trajectory	Lawn Mower	**Nominal Speed**	$1\,\mathrm{m\,s^{-1}}$
Simulation time sample	$0.05\,\mathrm{ms}$	**Vehicle Shape**	Circular
Noise deviation in position	$0.1\,\mathrm{m}$	**Noise deviation in orientation**	0.1rad
Current velocity in $\mathcal{R}_0$	$[-0.5\,\mathrm{m\,s^{-1}}, 0, 0]^\top$		
Conventional Flatness-based controller			
k_p	11.1	k_i	3.3
k_d	6.6	k_p^ψ	100
k_i^ψ	10	k_d^ψ	20
iPID			
α_x	1	α_y	1
K_P^x	11.1	K_p^y	11.1
K_d^x	6.6	K_D^y	6.6
k_p^ψ	100	k_i^ψ	10
k_d^ψ	20		

Table 4.3 – Simulation parameters - Hovercraft with sea current

In this experiment, a constant irrotational sea current and white measurement noise

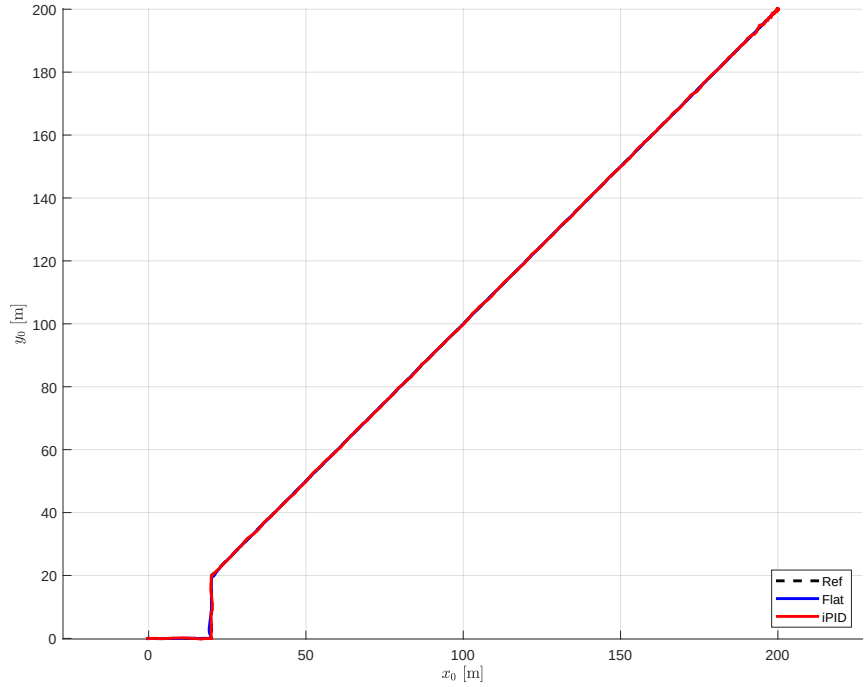


Figure 4.11 – Comparison of the trajectory of the flatness-based controllers with sea current - Blue: conventional flatness, Red: iPID, Black: Reference

are added. The current has a constant speed of $-0.5\,\mathrm{m\,s^{-1}}$ and is aligned with $\boldsymbol{x_0}$ axis of the inertial frame. The vehicle is therefore directly facing the current during the first segment, has it askew on the second segment and in diagonal all along the last segment.

The magnitude of the angle measurement is voluntary chosen to be much more consequential than the position measurement noise to emulate the difference in accuracy between a position sensor like a GPS and the angular measurement obtained with an IMU or a magnetic compass.

Figure 4.11 shows that the two controllers are robust to the constant current and noise on the three segments. As shown on figure 4.12, both error signals of the two vessels oscillate around 0 because of the measurement noise but are stable.

This experiment shows the behavior of the adaptive functions F_x and F_y of the iPID when the system is disturbed. Figure 4.13 shows that the the F_x function converges towards the estimated vales of the current velocity multiplied by the damping coefficient

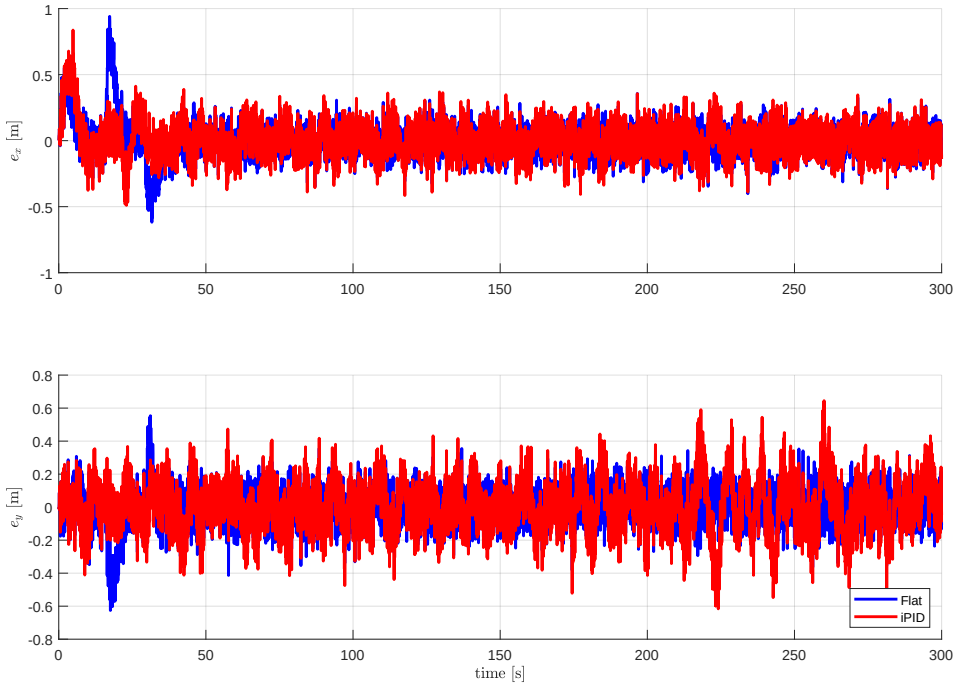


Figure 4.12 – Comparison of the tracking errors of the flatness-based controllers with sea current - a: $\boldsymbol{x_o}$ axis, b: $\boldsymbol{y_o}$ axis, Blue: conventional flatness, Red: iPID

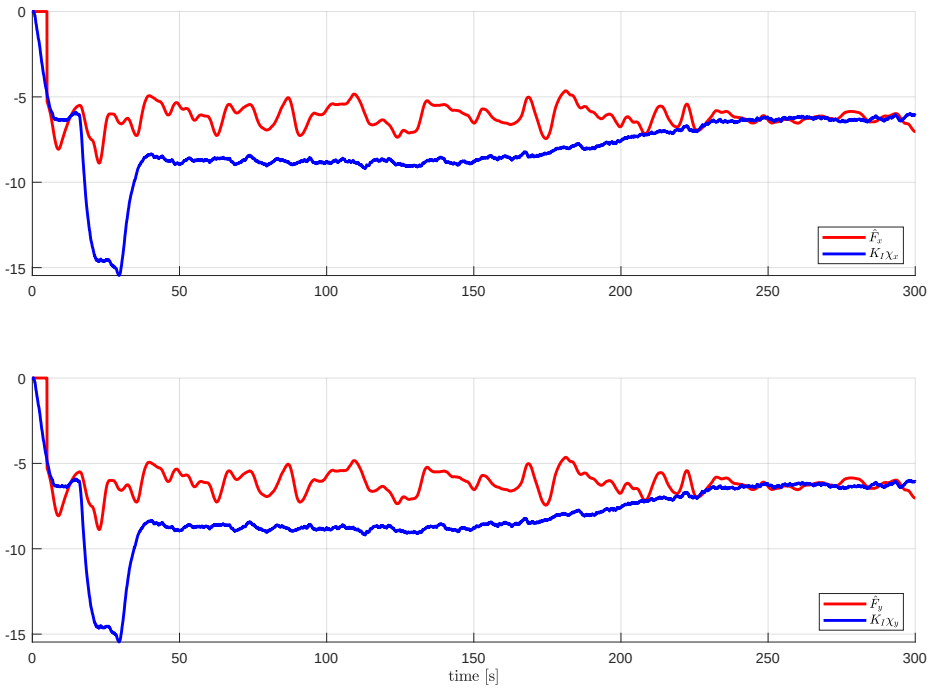


Figure 4.13 – Adaptive functions F_x and F_y on the hovercraft with constant sea current. Red: Adaptive functions iPID - Blue: Integral term conventional PID

$\widehat{F}_x \rightarrow -0.5\beta = -6$. In the conventional flatness-based controller, the integral term of the control law compensates the effects of the sea current. Figure 4.13 confirms that on simple case like this one, the adaptive functions of iPIDs have a similar behavior as the integral term of a conventional PID.

The experiments performed on the hovercrfat confirm the natural robustness of flatness-based controllers to both external disturbances and measurement noise.

4.5.2.3 Application to the surface vessel

In this section, the two controllers are applied to a generic surface vessel model with no modification of the control parameters. The same sea current and measurement noise are appied to the vehicle as in the previous section. The added mass and damping parameters of the vessel are given in table D.3. The surface vessel does not meet the hypothesis of circular shape of the hovercraft, the sway added mass and damping

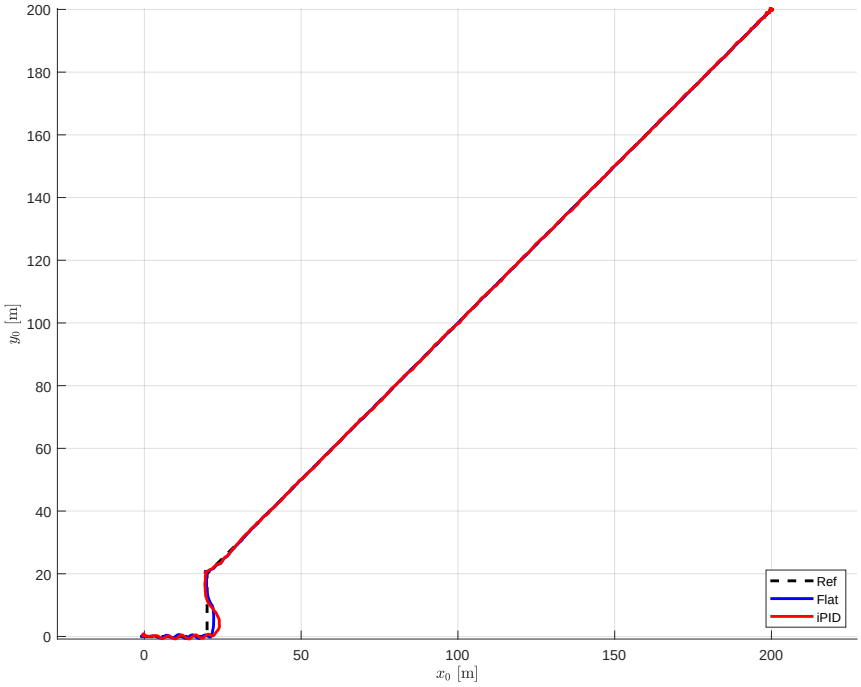


Figure 4.14 – Comparison of the trajectory of the flatness-based controllers on the SV with sea current - Blue: conventional flatness, Red: iPD, Black: Reference

parameters $Y_{\dot{v}}$ and Y_v are much larger than the surge counter-parts.

Figure 4.14 and figure 4.15 shows that both controllers still give good results when applied to the surface vessel in presence of current and noise. The natural robustness to model uncertainties of the flatness-based conventional controller is sufficient to keep the SV on track even though it is not meeting the hovercraft hypothesis at all.

4.5.2.4 Application to the surface vessel with initial error

As shown in the fully-actuated case in Section 4.5.1, the main weakness of flatness-based control (and feedforward linearization) is the behavior of the system when it gets too far from the desired trajectory. Feedforward linearization does not allow coming back on track if the vehicle exists a safety layer around the task. To test the behavior of the two controllers in such a case, they are evaluated on a single $\boldsymbol{x_o}$ axis segment with an initial error of $15\,\mathrm{m}$ on the $\boldsymbol{y_o}$ axis.

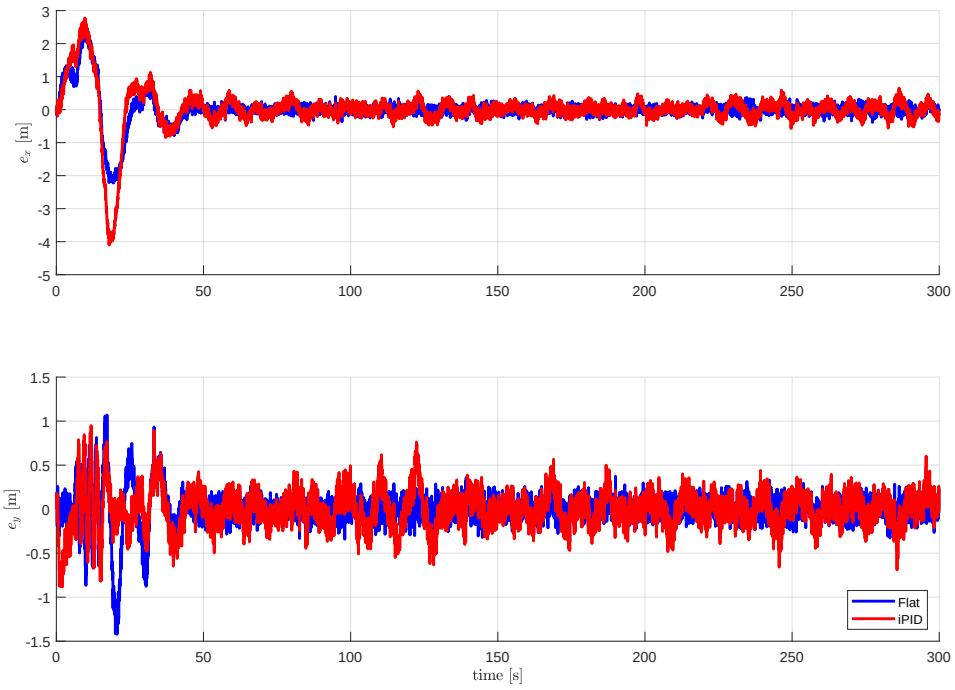


Figure 4.15 – Comparison of the tracking errors of the flatness-based controllers on the SV with sea current - a: $\boldsymbol{x}_\mathbf{o}$ axis, b: $\boldsymbol{y}_\mathbf{o}$ axis, Blue: conventional flatness, Red: iPD

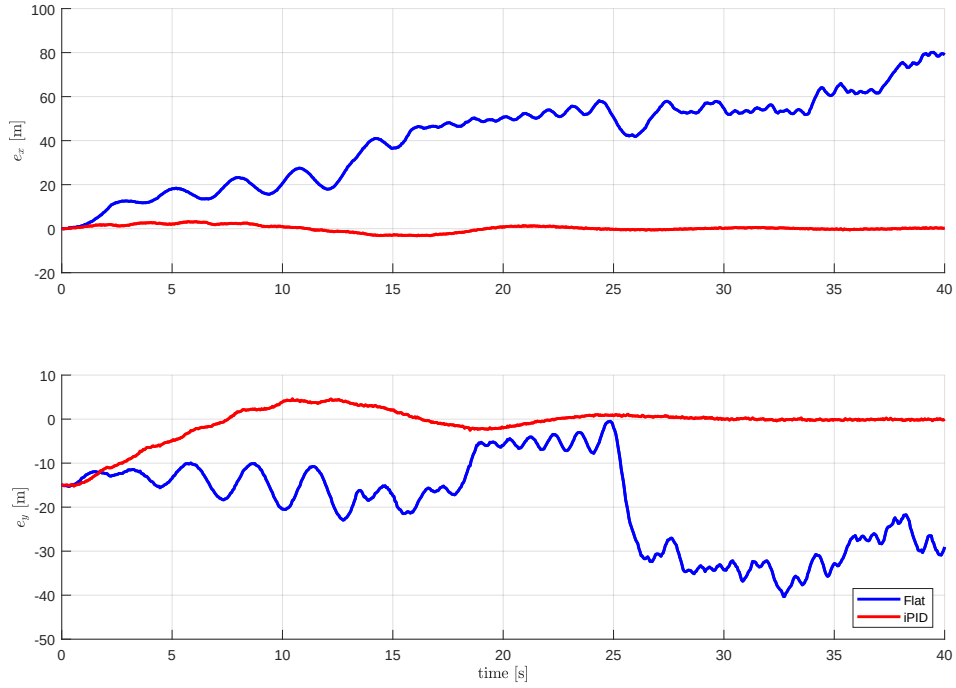


Figure 4.16 – Comparison of the tracking errors of the flatness-based controllers on the SV with a $15\,\mathrm{m}$ initial error - a: $\boldsymbol{x_o}$ axis, b: $\boldsymbol{y_o}$ axis, Blue: conventional flatness, Red: iPD

Figure 4.16 shows that the conventional flatness-based controller completely diverges from the trajectory with the initial error whereas the iPD allows converging back to the path.

Item	**Value**	**Item**	**Value**
Trajectory	Lawn Mower	**Nominal Speed**	$1\,\mathrm{m\,s^{-1}}$
Simulation time sample	$0.05\,\mathrm{ms}$	**Vehicle Shape**	Surface Vessel
Noise deviation in position	$0.1\,\mathrm{m}$	**in orientation**	0.1rad
Current velocity in $\mathcal{R}_0$	$[-0.5\,\mathrm{m\,s^{-1}}, 0, 0]^\top$		
Vehicle parameters			
m	$18\,\mathrm{kg}$	I_z	$0.61\,\mathrm{kg\,m}$
$X_{\dot{u}}$	2	$Y_{\dot{v}}$	50
$Y_{\dot{r}}$	2	$N_{\dot{r}}$	0.5
X_u	12.8	Y_v	100
Y_r	2	N_r	4
Conventional Flatness-based controller			
k_p	11.1	k_i	3.3
k_d	6.6	k_p^ψ	100
k_i^ψ	10	k_d^ψ	20
iPD			
α_x	1	α_y	1
K_P^x	11.1	K_P^y	11.1
K_D^x	6.6	K_D^y	6.6
k_p^ψ	100	k_i^ψ	10
k_d^ψ	20		

Table 4.4 – Simulation parameters - Surface Vessel with sea current

4.6 Partial conclusion on flatness-based control

This chapter applies flatness-based control to both fully-actuated and underactuated marine craft. It shows that flatness-based control is an interesting alternative for the control of sch vehicles as it is efficient an robust.

The flatness-based controller designed for underactuated systems naturally embeds a form of guidance as it allows turning position references into force and moment control inputs. In facts, the equations of flatness could also be used as a guidance principle associated with any controller as it notably allows calculating surge and yaw references for the vehicle to track.

Flatness-based control is also associated to intelligent PID to control underactuated surface vessels. Using the hovercraft hypothesis, a simplified flat model of the surface vessel is used to calculate a flatness-based controller. Then, an adaptive layer based on iPD is added to increase the robustness of the controller. The idea is to be able to successfully apply this controller to any surface vessel.

The simulation experiments performed on the hovercraft and surface vessel show that the flatness-based controller alone is already robust to model uncertainties, disturbance and measurement noise. Flatness also appears to be agood guidance solution as it allows calculating proper angular references for position trajectory tracking of an underactuated surface vessel. Yet, the last experiment performed on the surface vessel shows that conventional flatness-based control reaches its limit when the system is initialized far from the desired path. Nevertheless, this last experiment also shows that the association of an iPD layer increases the already robust flatness-based controller and compensate its weakness with great errors allowing the vehicle to converge back on the path after being initialized with a large position error.

Overall, flatness-based control is proven to be effective and robust for both fully-actuated and underactuated marine vehicles. Further work will be required to extend the results obtained on underactuated surface vessels to uneractuated AUVs. More studies on physical systems will be conducted.

CHAPTER 5

VECTOR THRUST FOR AUVS

Contents

This chapter studies effects of vector thrust on AUVs and proposes a new concept of thrust reconfiguration. Notably, it presents potential side-effects of conventional off-centered vector thrusters like those found in [43]. It shows that reconfigurable thrusters can have damaging coupled effects on AUVs and notably on smaller vehicles. Although it is physically bounded, the coupled forced effect created by off-center vector thrusters can jeopardize the control objectives of a vehicle.

The coupled effects of conventional vector thrusters are studied in details. Simulation studies show that they can be neglected on larger and heavier vehicles but not on smaller and lighter ones. More than the actual weight of the system, its shape can make it more or less naturally robust to these coupled effects making vector thrusters not viable for some systems.

To counteract these coupled effects, a new concept of vector thrust is introduced. With this concept, pairs of parallel thrusters are mounted on a ring rotating around the hull. Actuation of the ring allows directing the force of the pair of thrusters while keeping a decoupled moment or choosing the axis of the moment they generate keeping the force on a constant axis. A proof of concept has been designed: the PlaSMAR robot. Simulations show that the PlaSMAR robot has increased capabilities thanks to the reconfiguration ring.

5.1 Effects of vector thrusters

Vector thrust is one of the most common solution for increasing the maneuverability of a vehicle with a minimal amount of additional weight and expense. The first example of vector thrust in the marine context is the off-board engine of most private boats. The engine rotates around a vertical axis to generate thrust in the whole horizontal plane. Here, the rotation is often generated by the pilot or can be actuated.

For underwater vehicles, several technologies allow mimicking this principle in two or three dimensions [44, 43]. Underwater reconfigurable thrusters are usually composed of the main motor and one or two additional servos. Through mechanical contraptions, the servos actuate the rotation of the main motor or of the propeller axis to direct the thrust vector.

Vector thrusters generate off-center forces that can be used to control the orientation of the vehicle through their coupled moments. Yet, in general, only the moments generated by the actuators are taken into account both in the modeling and in the control of underactuated vessels. This section shows that the impact of the coupled force and moments on a vehicle depend on the physical properties of this vehicle. It also shows that even though the coupled force effects are considered as disturbances in general, they can also have an interest in applications like docking.

5.1.1 Coupled force effects

This section shows the coupled effects of off-centered vector thrusters and how they may interfere in the control calculations. The reconfigurable uqr-architecture introduced in section 1.2.2 is used as an example. A schematic of the vehicle is recalled in figure 5.1. The vehicle is equipped with a single 3-DOF stern thruster generating

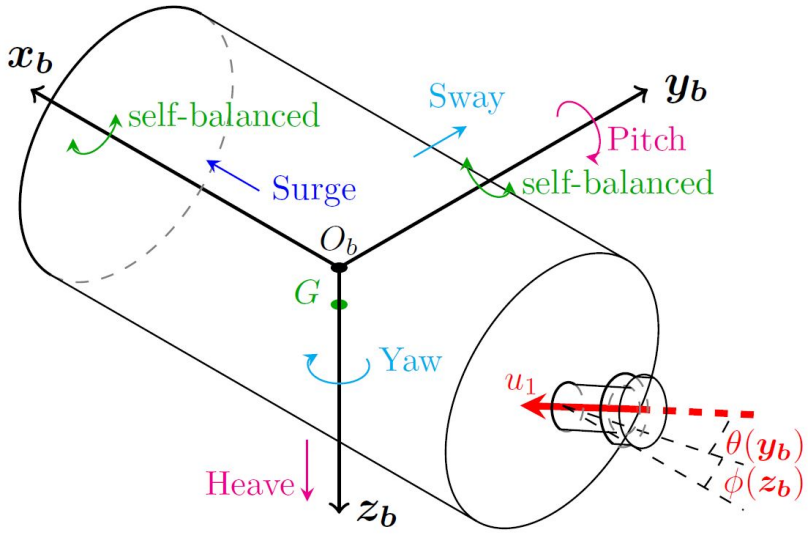


Figure 5.1 – Schematic of the example uqr-vehicle.

forces on the $\boldsymbol{x_B}$, $\boldsymbol{y_B}$ and $\boldsymbol{z_B}$ axes. As exposed in section 1.2.2, the vector of forces and moments created by the 1D3 thruster $\boldsymbol{\tau}$ is calculated using the TCM denoted $\boldsymbol{B}$:

$$\boldsymbol{\tau} = \boldsymbol{B}\boldsymbol{u} \tag{5.1}$$

where $\boldsymbol{u}$ is the vector of elementary forces created by the reconfigurable thruster. In this case, the 3-DOF thruster is equivalent to three fixed thrusters aligned with the $\boldsymbol{x_B}$, $\boldsymbol{y_B}$ and $\boldsymbol{z_B}$ axes respectively. The vector of forces is therefore $\boldsymbol{u} = [u_x, u_y, u_z]^\top$ with:

$$u_x = \cos(\theta_i)\cos(\psi_i)U \tag{5.2a}$$
$$u_y = \cos(\theta_i)\sin(\psi_i)U \tag{5.2b}$$
$$u_z = -\sin(\theta_i)U \tag{5.2c}$$

where θ_i and ψ_i are respectively the $\boldsymbol{y_B}$ and $\boldsymbol{z_B}$ reconfiguration angles of the vector thruster and $U = \sqrt{u_x^2 + u_y^2 + u_z^2}$ is the command thrust.

Considering that the force application point of the reconfigurable thruster is positioned at the very back of the hull of the vehicle and that the vector thruster rotates

around this point, the fixed-thruster equivalent TCM is:

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & -L \\ 0 & -L & 0 \end{bmatrix} \tag{5.3}$$

where L represents half of the vehicle hull length.

The vector of generated forces and moments is then of shape:

$$\boldsymbol{\tau} = \boldsymbol{B} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \\ u_z \\ 0 \\ -Lu_z \\ -Lu_y \end{bmatrix} \tag{5.4}$$

Equation (5.4) shows clearly the pairs of coupled forces and moments generated by the reconfigurable thruster. The sway force and yaw moment are coupled and the heave force and pitch moment are coupled.

From a controller perspective, one of the two coupled efforts is selected and controlled. Yet, the other coupled effort of each pair cannot be neglected as seen in the following.

5.1.2 Impacts on the dynamic model

The effects of coupled actuators are described on a simplified dynamic model representing an underwater vehicle equipped with a stern 3-D vector thruster. The compacted

model is:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{N}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{\tau} \tag{5.5}$$

where $\boldsymbol{N}(\boldsymbol{\nu}, \boldsymbol{\eta})$ represents the Coriolis and centripetal effects acting on the system as well as the damping, gravitational and buoyancy effects. $\boldsymbol{M}$ and $\boldsymbol{N}$ include added-mass effects.

Considering that the vehicle has a cylindrical shape (negligible off-diagonal added-mass parameters), the simplified dynamic model is given in component form:

$$m_1\dot{u} + n_1(\boldsymbol{\nu}, \boldsymbol{\eta}) = u_x \tag{5.6a}$$
$$m_2\dot{v} + n_2(\boldsymbol{\nu}, \boldsymbol{\eta}) = u_y \tag{5.6b}$$
$$m_3\dot{w} + n_3(\boldsymbol{\nu}, \boldsymbol{\eta}) = u_z \tag{5.6c}$$
$$m_4\dot{p} + n_4(\boldsymbol{\nu}, \boldsymbol{\eta}) = 0 \tag{5.6d}$$
$$m_5\dot{q} + n_5(\boldsymbol{\nu}, \boldsymbol{\eta}) = -Lu_z \tag{5.6e}$$
$$m_6\dot{r} + n_6(\boldsymbol{\nu}, \boldsymbol{\eta}) = -Lu_y \tag{5.6f}$$

where $m_1, ..., m_6$ are the diagonal coefficients of the mass and inertia matrix $\boldsymbol{M}$ and $n_1, ..., n_6$ are the components of the vector $\boldsymbol{N}$.

In this example, the pitch and yaw moments generated by the vector thruster are chosen as control inputs. The sway and heave forces generated by the vector thruster u_y and u_z appearing respectively in equation (5.6b) and equation (5.6c) are therefore considered disturbances.

Equation (5.6b) and (5.6c) show that the disturbing coupled forces are mainly mitigated by the mass and added mass parameters on the sway and heave axes m_2 and m_3. They are also counter-acted by the damping effects contained in n_2 and n_3:

$$\dot{v} = \frac{1}{m_2}(u_y - d_2 v - \bar{n}_2(\boldsymbol{\nu}, \boldsymbol{\eta})) \tag{5.7a}$$
$$\dot{w} = \frac{1}{m_3}(u_z - d_3 w - \bar{n}_3(\boldsymbol{\nu}, \boldsymbol{\eta})) \tag{5.7b}$$

where the linear damping terms $d_2 v$ and $d_3 w$ have been extracted from $n_2(\dot{\boldsymbol{\nu}}, \boldsymbol{\eta}) = \bar{n}_2(\boldsymbol{\nu}, \boldsymbol{\eta}) + d_2 v$ and $n_3(\dot{\boldsymbol{\nu}}, \boldsymbol{\eta}) = \bar{n}_3(\boldsymbol{\nu}, \boldsymbol{\eta}) + d_3 w$.

The impact of the disturbing sway and heave forces is therefore mitigated by the shape of the vehicle. The robustness of the natural sway and heave stability of the vehicle due to the shape of the hull and the mass distribution could allow neglecting

the sway and heave disturbing forces created by the vector thruster. Also, different hull shapes create different coupling terms in the mass and damping matrices counter-acting the coupled forces.

As an example, figure 5.2 shows the behavior of three cylindrical vehicles with different m_2 and m_3 mass parameters. The first vehicle is a close representation of the Remus100 (presented in details in section 3.5.2) (Blue lines on figure 5.2 and figure 5.3) , the second one (Red lines) has been artificially made ten times heavier on the sway and heave axes and the third one (Green lines) has been made ten times lighter on the same axes. The three vehicles are controlled in surge, pitch and yaw using the same $\mathcal{H}$-matrix based controller on the seabed scanning task. Note that, as shown in the next section, the construction of the $\mathcal{H}$ based controller as presented in chapter 3 indirectly assumes that the sway and heave disturbing forces are negligible. Nonetheless, as shown in this example, reasonably low sway and heave velocities are sufficient to ensure stability of the vehicle.

As expected, figure 5.3 shows that the coupled sway and heave forces generate less sway and heave velocity on the heaviest vehicle than on the two lighter ones. This natural stability results in better tracking performance as shown on figure 5.2 even though the lighter vehicle still shows reasonably good tracking performances too. It stays stable but oscillates around the desired trajectory.

Following the same idea, the $\mathcal{H}$-matrix controller has been applied to the RSM vehicle (presented in chapter 1), the Remus100 and a mock up of the ODIN. All three vehicles are equipped with a single rear 3-D vector thruster and tested on the seabed scanning task. Numerical values of the model coefficients for the three vehicles can be found in appendix D.

Figure 5.4 represents the position of the three vehicles. It shows that while the RSM (Blue line) and Remus100 (Red line) perform well on the task, the ODIN (Green line) oscillates around the desired trajectory. The spherical shape of ODIN does not offer a robust enough stability on the sway and heave axes to counteract the coupled effect of the vector thruster. While the vehicle oscillates in a stable tube around the desired trajectory, the tracking mission can hardly be considered successful.

This example goes to show that even though the hypothesis of negligible sway and heave disturbing forces is not formally correct, the natural stability of some systems is sufficient to counter-act the disturbing effects of the coupled forces. Note that, as demonstrated in [66, 64] and detailed in the following section, because the thrust of the

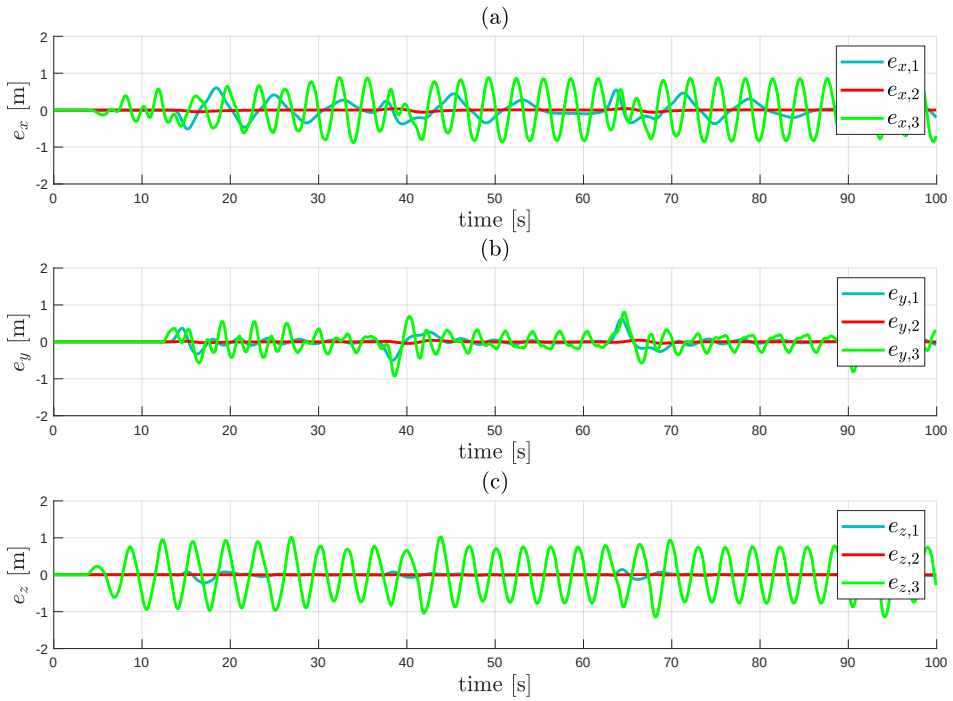


Figure 5.2 – Comparison of tracking errors of the 3 Remus-like vehicles - (a): $\boldsymbol{x}_\mathbf{o}$ axis, (b) $\boldsymbol{y}_\mathbf{o}$ axis, (c) $\boldsymbol{z}_\mathbf{o}$ axis - Blue: Normal, Red: Heavy, Green: Light

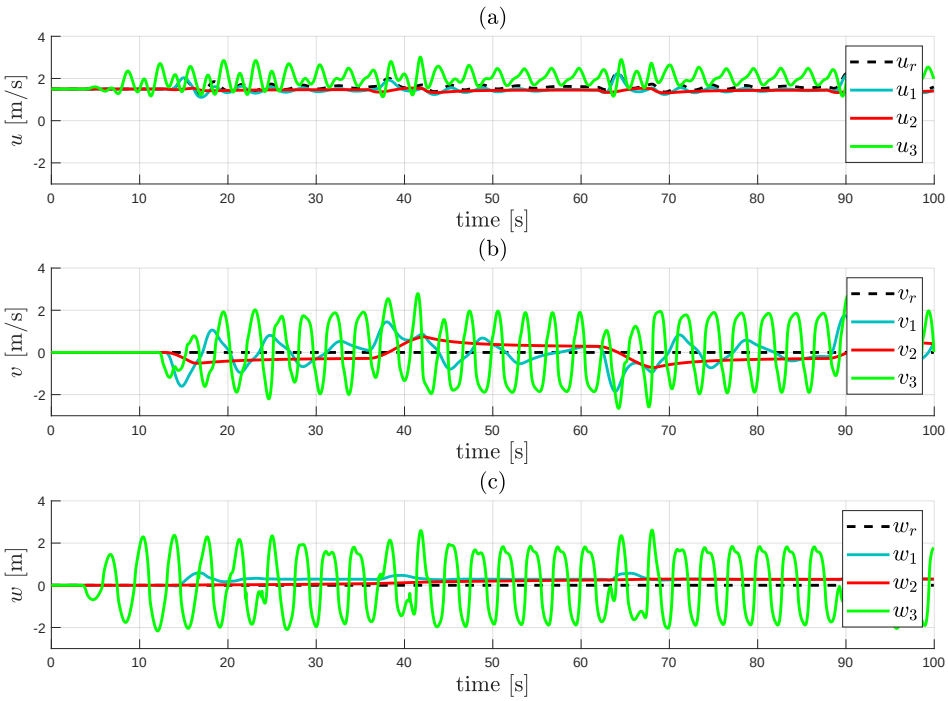


Figure 5.3 – Comparison of linear velocities of the 3 Remus-like vehicles - (a): $\boldsymbol{x_B}$ axis, (b) $\boldsymbol{y_B}$ axis, (c) $\boldsymbol{z_B}$ axis - Blue: Normal, Red: Heavy, Green: Light

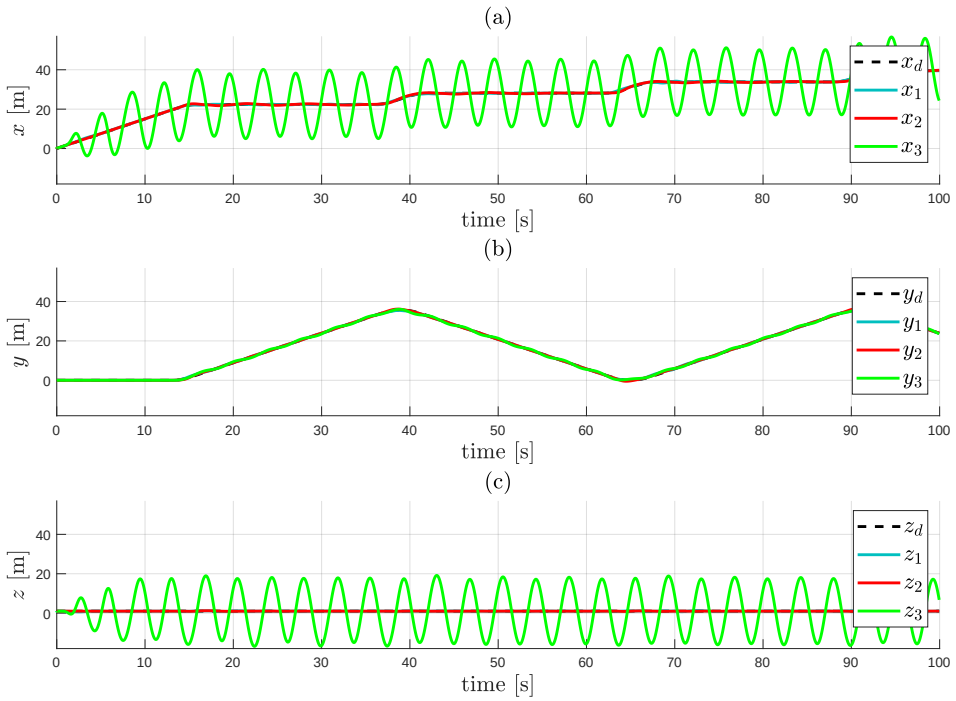


Figure 5.4 – Comparison of the position of the 3 vehicles on the Seabed Scanning task - (a): $\boldsymbol{x}_o$ axis, (b) $\boldsymbol{y}_o$ axis, (c) $\boldsymbol{z}_o$ axis - Blue: RSM, Red: Remus100, Green: ODIN

propeller is physically bounded, the disturbing forces are as well. Hence, the sway and heave speeds of the vehicle cannot completely diverge. Yet, from a control perspective, the sway and heave behavior of some systems can jeopardize the whole tracking mission as it is the case for ODIN in the last experiment.

5.1.3 Controller calculations for coupled actuators

Calculating the controller for a non-decoupled system is not trivial. As shown in the previous section and in appendix C, the coupled effects of the input can disturb both the controlled and uncontrolled DOF of the system to a point where the tracking task cannot be considered successful. Some of the controllers introduced in chapter 2 require the assumption of negligible coupled force when the moment is used for control of the position of the vehicle through kinematic or dynamic coupling or any other guidance principle. This is notably the case for the sliding mode underactuated controller described in [40] and used in simulations in section 3.4 and for the $\mathcal{H}$ based controller introduced in chapter 3.

In the case of a vehicle equipped with one stern vector thruster, the underactuated SMC controller would use the first order derivative of the sway and heave dynamical equations to exploit the dynamic couplings of the system and control the sway and heave modes with the yaw and pitch inputs respectively. Considering a simplified model of an underwater vehicle (diagonal mass and damping matrices, linear damping), the sway, heave, pitch and yaw equations of the dynamic model taking the coupled forces into account are:

$$\dot{v} = \frac{1}{m - Y_{\dot{v}}} \left(u_y + (m - X_{\dot{u}})ur - (m - Z_{\dot{w}})wp + Y_v v\right) \tag{5.8a}$$

$$\dot{w} = \frac{1}{m - Z_{\dot{w}}} \left(u_z + (m - Y_{\dot{v}})vp - (m - X_{\dot{u}})uq + Z_w w\right) \tag{5.8b}$$

$$\dot{q} = \frac{1}{I_y - M_{\dot{q}}} \left(-Lu_z + (Z_{\dot{w}} - X_{\dot{u}})uw + (I_x - I_z + N_{\dot{r}} - K_{\dot{p}})pr + M_q q\right) \tag{5.8c}$$

$$\dot{r} = \frac{1}{I_z - N_{\dot{r}}} \left(-Lu_y + (X_{\dot{u}} - Y_{\dot{v}})uv + (I_y - I_x + K_{\dot{p}} - M_{\dot{q}})pq + N_r r\right) \tag{5.8d}$$

Taking the example of the sway equation (5.8a), the second order time derivative of

v gives:

$$\begin{aligned}\ddot{v} &= \frac{1}{m - Y_{\dot{v}}}\left(\dot{u}_y + (m - X_{\dot{u}})(\dot{u}r + u\dot{r}) - (m - Z_{\dot{w}})(\dot{w}p + w\dot{p}) + Y_v\dot{v}\right) \\ &= \frac{1}{m - Y_{\dot{v}}}\left[\dot{u}_y + (m - X_{\dot{u}})(\dot{u}r + \frac{u}{I_z - N_{\dot{r}}}\left(-Lu_y + (X_{\dot{u}} - Y_{\dot{v}})uv + N_r r\right)\right) \\ &\quad -(m - Z_{\dot{w}})(\dot{w}p + w\dot{p}) + Y_v\dot{v}\Big] \end{aligned} \tag{5.9}$$

where both u_y and $\dot{u}_y$ appear due to the coupled force effect of the vector thruster. The sway dynamics of the vehicle are now expressed as a generalized system: the first order derivative of the input appears in the equation(see [47] and appendix B for more details). Hence, the strategy used to establish the SMC on the decoupled system cannot be used as such in this case.

In the same way, the $\mathcal{H}$ matrix based controller introduced in chapter 3 indirectly assumes that the coupled force effects on the sway and heave are negligible. This time, it is in facts the sway and heave control speeds that are assumed to be 0 in the kinematic stage of the controller to establish the relation $\boldsymbol{\nu}_c = \mathcal{H}\boldsymbol{T}^{-1}\boldsymbol{J}(\boldsymbol{\eta})^{-1}\dot{\tilde{\boldsymbol{\eta}}}_{\boldsymbol{E}}$, where $\dot{\tilde{\boldsymbol{\eta}}}_{\boldsymbol{E}}$ is the sum of the desired velocity and the PI control law. Without assuming $v_c = w_c = 0$, the kinematic relations $v_c = \bar{v}_E - \varepsilon_x r_c$ and $w_c = \bar{w}_E + \varepsilon_x q_c$ cannot be used to cancel the sway and heave errors using the yaw and pitch speeds respectively as they were in section 3.3.

Nonetheless, some controller designs can take the coupling effects of vector thrusters into account. This is the case of the adaptive and integral LOS guidance solutions of [61, 62] described in section 2.1. In these controllers, the sway force (and heave force if applied to underwater vehicles) is not directly taken into account but is instead compensated through the calculations of the adaptive crab angle. Using the adaptive crab angle in the LOS calculations allows considering the surge and sway speeds ratio and couteracting the disturbing sway speed. The coupled effect is therefore estimated as a disturbance and compensated in the guidance calculations.

Also, the work of [66] introduces another controller taking the coupled force effects of an underactuated surface vessel equipped with a vector thruster into account. The controller is derived using the back stepping technique and Lyapunov theory. It includes the uncontrolled sway modes and shows that the system is stable even though it is disturbed by the coupled forces. In facts, this work shows that because the inputs of the system are physically bounded by the maximum thrust of the motor, the system is

stable.

Some more theoretical work like [115] demonstrate the mathematical stability of coupled cascaded systems like the the AUV with vector thrusters using the property of *forward completeness* (see [5] for in depth studies of forward completeness). Yet, the mathematical stability of the system demonstrated in this work does not ensure tracking performances. Even though the system cannot be completely destabilized, the uncontrolled sway modes disturb the trajectory tracking mission. The use of reconfigurable thrust is to consider for naturally stable and robust systems but is not ideal for lighter vehicles.

5.1.4 Force control with vector thrusters

On of the interest of vector thrust is the possibility to control either the forces generated by the thruster or the coupled moments. As depicted before in this section and all along this work, the traditional way of controlling marine craft is to use the moment generated by an off-center motor and propeller or control surface to control the steering of the vehicle. Doing so, the force component is either neglected or considered a disturbance. In this section, simulation experiments performed on the Remus100 vehicle equipped with a single 3-D reconfigurable thruster show that the sway and heave force components generate by the thruster can be used directly in the controller in specific use cases like docking and station keeping.

Of course, intuition and experience show that controlling a free body in water with rear forces is unstable, the vehicle would turn around so that the forces are actually pulling it instead of pushing. This instability is due to the hydrostatic and hydrodynamic effects acting on the hull of the craft. Consequently, in this section, the reconfigurable thruster is positioned at the front of the vehicle. This is equivalent to having a vehicle going “backwards” with a tracking point positioned at the rear.

One major issue when “pulling” the vehicle is that the water flow generated by the thruster is directly hitting the hull. This is energetically inefficient and potentially damaging for the vehicle in the long term. It must therefore be limited to short range applications. One noteworthy example where force control is most useful and more efficient that moment control is the docking maneuver.

The docking maneuver is a really complex case study for the guidance and control of autonomous marine craft that has been solved in many different ways like conventional

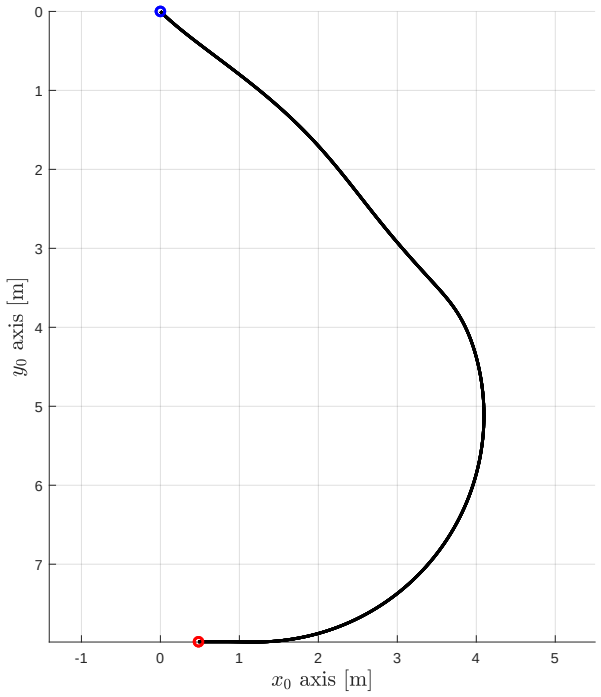


Figure 5.5 – Top view of the desired docking trajectory in the inertial frame

control or machine learning [119, 144]. Here, a very simplified docking maneuver is presented. It is similar to the maneuver performed in ports with boats. A virtual dock parallel to the $\boldsymbol{x_0}$ axis is simulated and the vehicle must approach the dock, turn to align with it and then maintain its position alongside the virtual dock. The task is composed of one approach segment at a $135°$ angle to the dock, a full turn to align the vessel with the dock and a station keeping subtask when the vessel reaches the virtual dock. The desired docking trajectory is displayed on figure 5.5. The blue circle represents the initial position of the vehicle where the docking maneuver is initialized and the red dot represents the end position. The vehicle takes around 10 seconds to reach the red dot and then maintains its position on the red dot for as long as the simulation lasts. The virtual dock is materialized by the horizontal axis on figure 5.5.

The position of the tracking point of the uqr-vehicle is controlled using the $\mathcal{H}$-based controller with the corresponding forces generated by the thruster. No compensation mechanism is necessary so the $\mathcal{H}$ matrix in this case is the identity. The controller is

almost equivalent to a single stage PID controller defined in the inertial frame (see Chapter 3 for more details).

The attitude of the vehicle is left uncontrolled. The hydrodynamic effects acting on the hull naturally align the vehicle to the water flow surrounding it. Hence, with no current the vehicle aligns with the velocity vector at the tracking point. With currents, the vehicle finds an equilibrium between the direction of the velocity and the direction of the current. In the docking maneuver, the turn before station keeping aligns the vessel with the virtual dock. In the presence of sea current, the vehicle naturally aligns with the current field when station keeping.

The docking experiment is performed with a constant sea current of $-0.5\,\mathrm{m\,s^{-1}}$ velocity aligned with the $\boldsymbol{y_0}$. Two control strategies are compared on the Remus100 equipped with a rear reconfigurable thruster. The position of the first vehicle is controlled conventionally, using the moments generated by the thruster. Kinematic couplings are used in the controller to compensate the lack of sway and heave actuation. The second vehicle is controlled directly with the force generated by the thruster. No compensations mechanism is involved. The second vehicle is initiated backwards on the approach segment for stability.

Recalling the formalism introduced before, the pair of DOF vectors in the first case is:

$$\boldsymbol{h_{O_B}}^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \boldsymbol{h_E} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.10}$$

while in the second case with force control it is

$$\boldsymbol{h_{O_B}}^2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{h_E} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.11}$$

The first vehicle is considered to be actuated in surge, pitch and yaw and the three position coordinates of the tracking point are controlled through a guidance mechanism (in this example based on the $\mathcal{H}$ matrix). The second vehicle is considered to be actuated in surge, sway and heave and the three position coordinates of the tracking point are controlled. This second case does not require any guidance or compensation mechanism.

The position tracking errors of the two vehicles are displayed on figure 5.6. The force-controlled vehicle (Red line) appears much more stable than the moment-controlled one (Blue lines). Notably, figure 5.6 shows that force control allows much better station keeping for this vehicle. In moment-control, the vehicle circles around the final point in a stable ellipsoid as seen on figure 5.7.

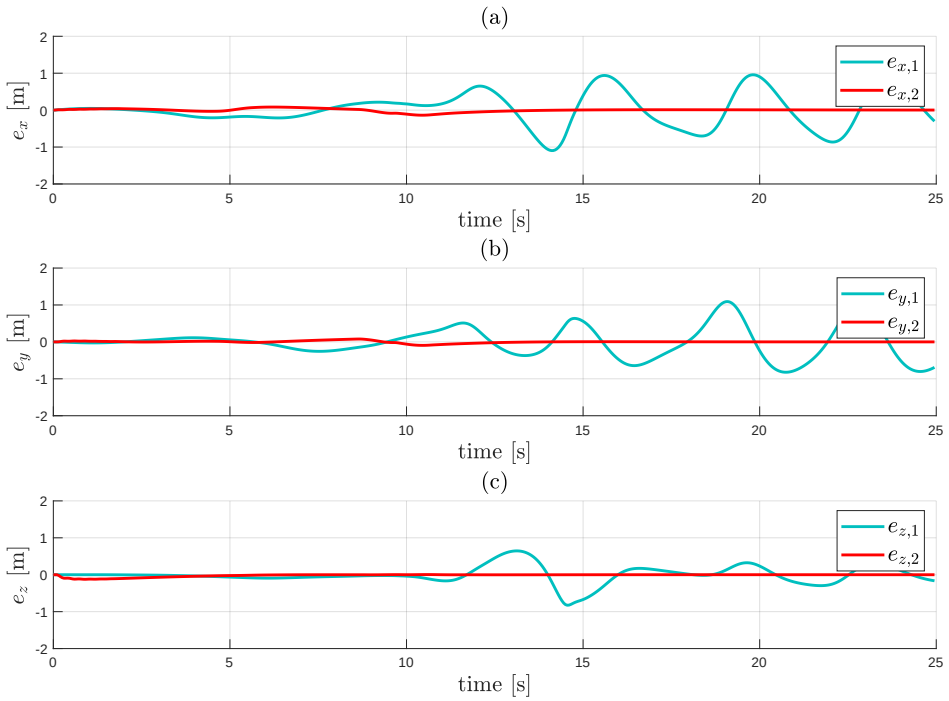


Figure 5.6 – Comparison of position tracking errors on the docking task - a: $\boldsymbol{x}_\mathbf{o}$ axis, b: $\boldsymbol{y}_\mathbf{o}$ axis, c: $\boldsymbol{z}_\mathbf{o}$ axis - Blue: Moment control, Red: Force control

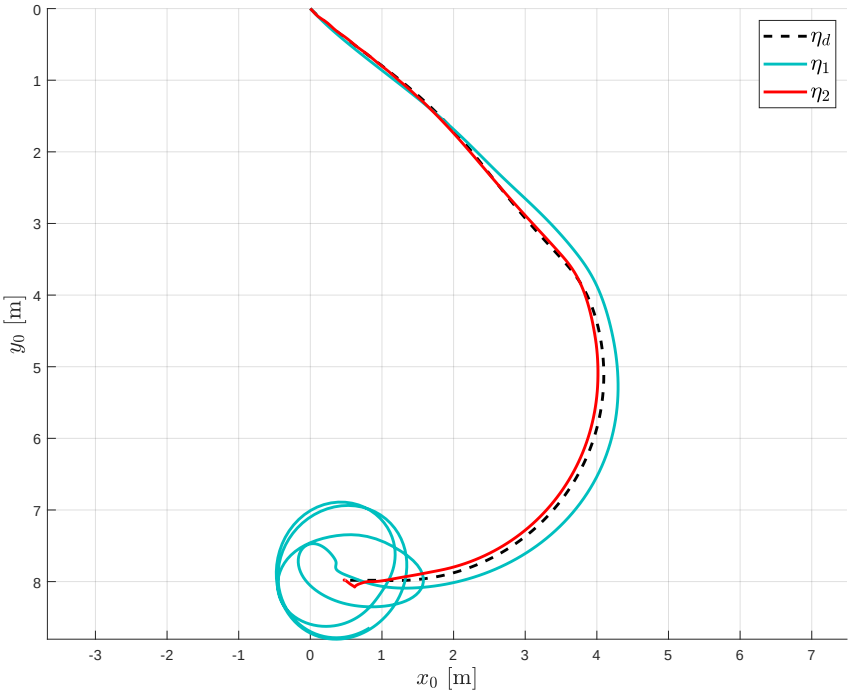


Figure 5.7 – Top view of the trajectory of the two vehicles on the docking task - Blue: Moment control, Red: Force control

5.2 New vector thrust concept: *PlaSMAR*

This section introduces a new concept of reconfigurable thrust. The idea is to direct and reposition a pair of thrusters to allow generating a directed force and a decoupled moment. This concept is notably suited for cylindrical-hull vehicles as it relies on a ring carrying the thrusters around the hull.

It has been developed on the test platform *PlaSMAR*[1]. PlaSMAR is an easily reconfigurable robot designed to easily emulate any fixed-thrusters arrangement as well as to be equipped with the reconfiguration ring. The first version of this vehicle can be seen on figure 5.8 in a four fixed-thrusters configuration equivalent to the *RSM* vehicle.

This first prototype has been mainly 3D printed. It is inspired by the *Blue ROV* (Blue Robotics company) and by the printable AUV designed in [146]. On figure 5.8, the vehicle is equipped with two 3D printed black rings holding two thrusters each. The fixed rings allow positioning and orienting pairs of thrusters to reproduce commercial propulsive arrangements and facilitate testing fixed configurations.

This section details the concept of the rotating ring used for reconfiguration of a pair of parallel thrusters. Notably, it presents the thrust allocation used on two configurations of PlaSMAR and control solutions.

5.2.1 Reconfiguration ring

Conventional thruster reconfiguration is pushed a step further with the PlaSMAR concept. The idea is to direct a pair of parallel thrusters to control both the force and the moment they generate without introducing new coupling terms. To do so, two parallel thrusters are mounted on a ring rotating around the hull of the vehicle. They can be mounted either tangent to the ring or parallel to its axis of rotation as seen on figure 5.9.

The first configuration of figure 5.9 allows directing the summed force of the two thrusters in the plane of the ring while maintaining the moment actuation around the ring axis created by differential thrust. In the second configuration, the force of the two thruster is kept parallel to the axis of rotation of the ring but the axis of the moment they generate can be directed.

Of course, in order to equip the ring on a vehicle, it must have a cylindrical hull or at least a cylindrical section of hull around which the ring can rotate. On a cylindrical-hull

1. Plateforme Sous-Marine Autonome Reconfigurable

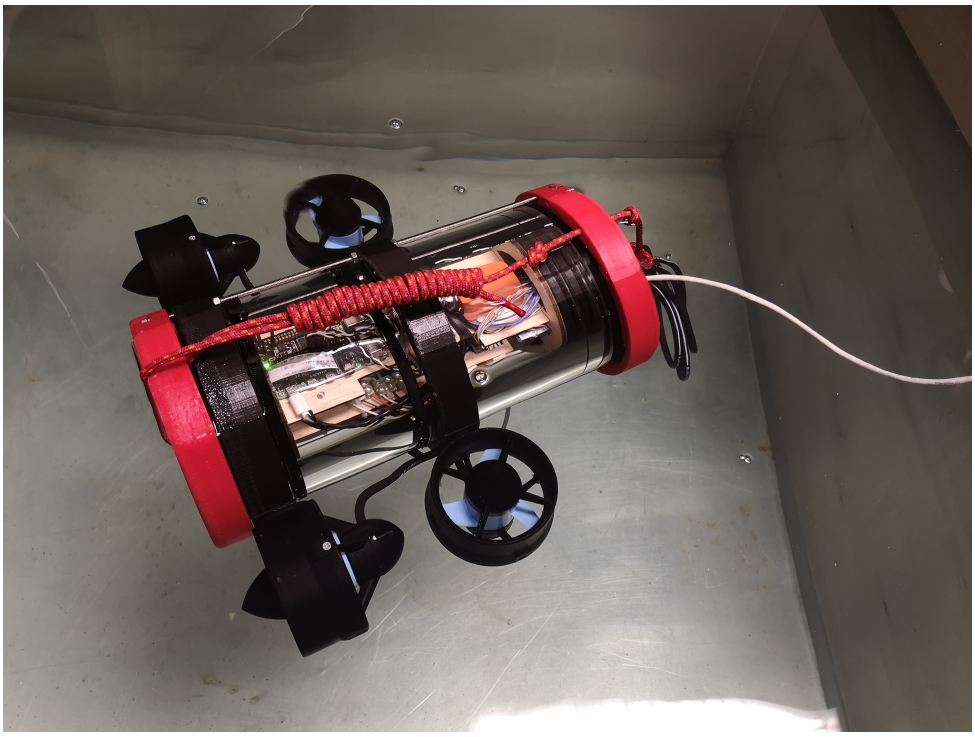
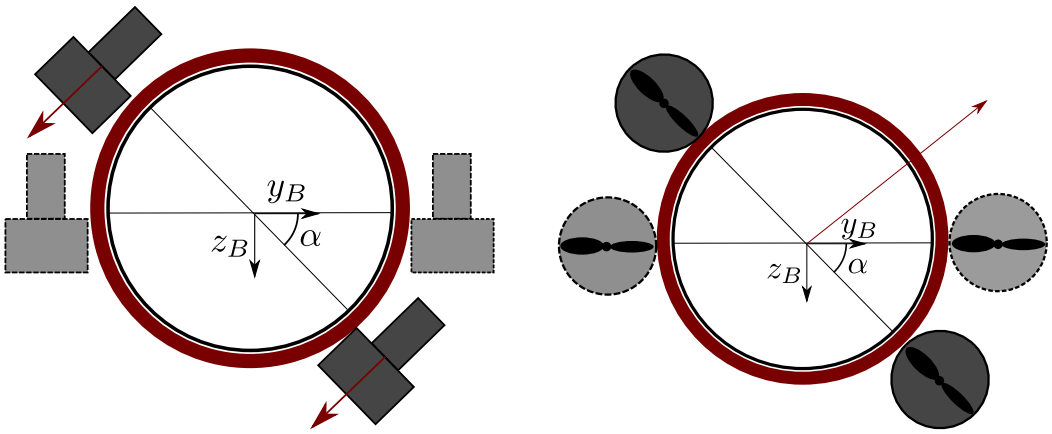


Figure 5.8 – Picture of PlaSMAR during one of the first pool tests. Configuration equivalent to *RSM* with four fixed thrusters.



(a) First configuration. Red arrows: thrust vectors of each thruster

(b) Second configuration. Red arrow: axis of the generated moment

Figure 5.9 – Schematics of the rotating ring principle. Black ring: hull of the AUV, Red ring: Rotating ring

(a) $0°$ reconfiguration angle (b) $90°$ reconfiguration angle

Figure 5.10 – 3-D renders of the first configuration of PlaSMAR. The rotating ring is placed at the front (rendered on Fusion360)

(a) $0°$ reconfiguration angle (b) $90°$ reconfiguration angle

Figure 5.11 – 3-D renders of the second configuration of PlaSMAR. The rotating ring is placed at the rear (rendered on Fusion360)

vehicle, the ring can be positioned anywhere along the hull and in the $(\boldsymbol{y_B}, \boldsymbol{z_B})$ plane. It rotates around the $\boldsymbol{x_B}$ axis of the cylinder. The two configurations of figure 5.9 are displayed on 3-D renders of PlaSMAR on figure 5.10 and figure 5.11 respectively.

Figure 5.10 and figure 5.11 shows two stages of the design of PlaSMAR. Initially, as shown on **??**, four metallic bars were used to hold the different rings around the hull. The bars were only pushed inside the two black end pieces holding on the front and back flanges. The latest version shown on **??** uses two threaded rods to hold the two end pieces appearing in red together. The ring are mounted on two points and secured with nuts. The rotating rings presented on figure 5.10 and figure 5.11 are simplified proof of concepts of the system. The motor is bigger than needed and should be encased. Also, the front ring on both configurations should be in the middle of the hull to avoid

coupled pitch effects as it is on figure 5.8.

In the first configuration(figure 5.10), the two front thrusters are mounted on the reconfiguration ring. They generate a force in the $(\boldsymbol{y_B}, \boldsymbol{z_B})$ plane parameterized by the rotation angle of the ring and a roll moment independent from their configuration. This contraption therefore allows actuation in sway and heave whilst maintaining decoupled roll. With the ring in its neutral $0°$ configuration (figure 5.10a), the thrusters are aligned with the $\boldsymbol{z_B}$ axis and generate a pure heave force. At a 90 degrees angle (figure 5.10b), the thrusters are aligned with the $\boldsymbol{y_B}$ axis and generate a pure sway force. In between the two extreme configurations, they can generate any diagonal force in the $(\boldsymbol{y_B}, \boldsymbol{z_B})$ plane. With the two rear fixed-thrusters, this vehicle is actuated in surge, sway, heave, roll and yaw. Only the pitch moment is not controlled.

In the second configuration (figure 5.11), the two back thrusters are mounted on the reconfiguration ring. The two thrusters are parallel to the $\boldsymbol{x_B}$ of the vehicle so they generate a surge force in any configuration of the ring. The main interest of the reconfiguration ring in this case is the possibility to direct the axis of the moment generated by the two thrusters. At a 0 degree angle (figure 5.11a), the thrusters are in the $(\boldsymbol{x_B}, \boldsymbol{y_B})$ plane aligned with $\boldsymbol{x_B}$ and generate a yaw moment while at a 90 degrees angle (figure 5.11b), they are in the $(\boldsymbol{x_B}, \boldsymbol{z_B})$ plane and generate a pitch moment. In the intermediary configurations, they can generate a moment around any axis of the vertical plane $(\boldsymbol{y_B}, \boldsymbol{z_B})$. With the two front fixed-thrusters, this vehicle is actuated in surge, heave, roll, pitch and yaw. Note that the rear rotating ring alone constitute an alternative to the common uqr-vessels. The two rear thrusters mounted on the rotating ring can independently generate a surge force, a pitch and a yaw moment. With the proper guidance, these three DOF are sufficient to perform any underwater position tracking task. This solution is a viable alternative to the usual fixed-thruster and control surfaces actuation of torpedo-shaped vehicles.

Overall, a pair of thrusters mounted on the reconfiguration can be utilized to generate an adjustable force vector whilst maintaining the moment or to direct the axis of the moment whilst maintaining the force. Effectively, the reconfiguration ring creates a new actuated DOF on the vehicle.

Several solutions can be designed to actuate the ring. The first and most simple one is to use a small servomotor with gear reduction placed on the outside of the hull. This is by far the simplest solution mechanically but it requires water proof casing to protect the electronic components exposed to water. Also, adding components on the

hull modifies its hydrodynamic profile and can be detrimental. It could also be difficult to fix the motor assembly to the hull without risks of leakage. A second solution could be to magnetically connect the outside ring to another ring inside the hull holding magnets and actuate the inside ring. This way, no external components are needed. This kind of magnetic contraption has been used for example in [168, 73, 44] to transmit force through the hull of a craft with no hole. Finally, and notably in the first configuration presented on figure 5.10a, the thrusters themselves could be used to rotate the ring. Unlatching the ring from the hull, the roll moment generated by the two thrusters of the first configuration would rotate the ring. A mechanical latch on the outside would be necessary or one could imagine some form of magnetic latching from the inside. This solution implies that the thursters are reconfigured at times when no roll is needed in the application since the ring will be rotating freely around the hull.

One of the other interest of the reconfiguration ring is to compensate roll disturbances. In facts, section 3.4.1 shows that roll actuation can be very useful in association with heave to compensate for the lack of sway. Yet, this example shows one of the main problem of staying at a non-zero roll angle. Because most vehicles are design to operate at a zero degree roll angle and because task are usually defined in the inertial frame, a non-zero roll angle equilibrium can jeopardize the actuation of another DOF of the task. In the example of section 3.4.1, the vehicle is rolling on the side to compensate for sway but doing so, the two rear thrusters responsible for heave are misaligned with the vertical axis of the inertial frame. At the extreme, the vehicle is not able to rotate around the vertical axis of the inertial frame anymore because a 90 degrees roll angle puts the two rear thrusters in the vertical plane of the inertial frame. The vehicle looses a DOF with respect to the task (rotation around the vertical axis of $\mathcal{R}_0$). Therefore, the reconfiguration ring could be used to compensate for this phenomena and maintain the rear thrusters in the horizontal plane of $\mathcal{R}_0$. Using the ring to rotate of the opposite of the roll angle of the vehicle, the thrusters mounted on the ring can be kept fixed in the horizontal plane of the inertial frame. This behavior can be used to compensate disturbances in roll as well as consequences of sway compensation using roll.

Then, in the second configuration presented on figure 5.10b, compensating the roll angle would allow the two rear thrusters to generate a moment around the vertical axis of the inertial frame independently from the attitude of the vehicle (considering that pitch is different from $90°$). In this use case, the reconfiguration ring does not really add a new DOF to the vehicle but makes it more robust to roll disturbance.

The roll compensation behavior is not as useful in the first configuration of figure 5.10 because roll itself is actuated and can be stabilized. Yet, one could imagine scenarios where the vehicle needs to dives vertically whilst suffering from roll disturbances. In this case, it would be useful to be able to keep the two thrusters vertical w.r.t. the inertial frame to maintain full heave capacity.

5.2.2 Thrust allocation of PlaSMAR

Considering the first configuration of PlaSMAR figure 5.10a, this section exposes the thrust allocation of the reconfigurable ring. In this case the front thrusters are mounted on a ring placed around the cylindrical hull of the vehicle, approximately in the middle of its length. In section 1.2.1 and section 1.2.2, thrust allocation is performed using the TCM matrix denoted $\boldsymbol{B}$. The TCM is used to calculate the necessary thrust of all the thrusters of the vehicles for a given command vector $\boldsymbol{\tau}_c$:

$$\boldsymbol{\tau}_c = \boldsymbol{B}\boldsymbol{u} \tag{5.12}$$

where $\boldsymbol{u}$ is the vector containing all the elementary thrusts. The relation (5.12) needs to be inverted to perform thrust allocation.

For fixed thrusters, $\boldsymbol{B}$ is constant. In the fully-actuated case without redundancy, $\boldsymbol{B}$ is invertible and thrust allocation is performed using:

$$\boldsymbol{u} = \boldsymbol{B}^{-1}\boldsymbol{\tau}_c \tag{5.13}$$

In the underactuated case, two methods are detailed in section 1.2.1: the Moore-Penrose pseudo-inverse used notably for decoupled propulsive arrangement and the space reduction which consists in inverting $\boldsymbol{B}$ in a reduced subspace where the vehicle can be considered fully-actuated.

For vector thrusters, the same methods are used considering an equivalent TCM representing a virtual fixed-thruster equivalent of the reconfigurable arrangement [30]. Then, a 3-DOF vector thruster is represented with 3 orthogonal fixed virtual thrusters and the TCM is calculated considering the virtual thrusters. The main interest of this solution is that it provides a constant TCM independent from the reconfiguration angles of the vector thrusters. The angles and thrust are reconstructed after thrust allocation is performed on the fixed-thrusters equivalent.

Here, with the thrusters mounted on the ring their position is not constant. It is therefore difficult to find a fixed-thrusters equivalent. In the first configuration of PlaSMAR, the two front thrusters are used to generate a force in the $(\boldsymbol{y_B}, \boldsymbol{z_B})$ plane and a roll moment without coupling. One fixed-thrusters solution emulating the generation of a force in the $(\boldsymbol{y_B}, \boldsymbol{z_B})$ plane is to consider two orthogonal thrusters placed at the center of the hull. Yet, this solution does not allow generating roll moment. One or several additional virtual thrusters would be necessary to generate roll which would modify the dimension of the problem. A configuration with three thrusters placed in the vertical plane all around the hull with $120°$ angle one with the other could be used to generate sway, heave and yaw.

In a first approximation, the ring is considered to have a perfect tracking of the commanded angle $\alpha = \alpha_c$. Therefore, the TCM can be considered locally constant to perform thrust allocation. The controller outputs a control effort vector and notably the sway and heave forces as well as the roll moment, Y_c, Z_c and K_c respectively. The control angle of the ring α_c is calculated geometrically for a given pair Y_c, Z_c:

$$\alpha_c = \arctan_2(Z_c, Y_c) \tag{5.14}$$

The $\arctan_2$ function is used for a complete four quadrants definition of the angle α_c. More studies are needed to establish the optimal solution of α_c considering the rotation speed of the ring and the variation rate of the thrusters. Practically, if the thrusters are considered to be able to switch direction and to vary its thrust fast enough, α could be limited to a $[0, \pi/2]$ interval and cover all possible forces of the plane. For now, the thrust allocation procedure favors rotating the ring over inverting the thrust of the two thrusters. As seen in the following simulation, this can result in a light oscillation due to numerical delay.

The TCM can then be calculated for a given α_c as:

$$\boldsymbol{B}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & \sin(\alpha_c) & \sin(\alpha_c) \\ 0 & 0 & -\cos(\alpha_c) & -\cos(\alpha_c) \\ 0 & 0 & -R & R \\ 0 & 0 & 0 & 0 \\ -R & R & 0 & 0 \end{bmatrix} \tag{5.15}$$

where R is the radius of the hull. The two first columns correspond to the two rear thrusters actuating surge and yaw and columns three and four correspond to the two thrusters mounted on the ring. Equation (5.15) confirms that when the ring angle α_c is 0 degrees, the two front thrusters generate a pure heave force and when the angle is 90 degrees, they generate a pure sway force. Then, between 0 and 90 degrees, they generate a diagonal force in the $(\boldsymbol{y_B}, \boldsymbol{z_B})$ plane. Equation (5.15) also shows that the roll moment generated by the two thrusters mounted on the ring is independent from the angle of the ring and is decoupled from the sway and heave forces.

Because the sway and heave modes are coupled with α_c different from 0 or 90 degrees, space reduction is not suited for inverting $\boldsymbol{B}_1$. The Moore-Penrose pseudo-inverse in used in this case (*see* **??**. Simulation results presented in the following show that the Moore-Penrose pseudo-inverse behaves properly in this case but this result is hardly general. More investigation on other ineverting methods based on quadratic programming would be needed to ensure that the inversion of $\boldsymbol{B}_1$ is always successful. The Moore-Penrose pseudo-inverse of the TCM is:

$$\boldsymbol{B}_1^{\dagger} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2R} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2R} \\ 0 & \frac{\sin(\alpha_c)}{4} & -\frac{\cos(\alpha_c)}{4} & -\frac{1}{2R} & 0 & 0 \\ 0 & \frac{\sin(\alpha_c)}{4} & -\frac{\cos(\alpha_c)}{4} & \frac{1}{2R} & 0 & 0 \end{bmatrix} \tag{5.16}$$

A more rigorous approach would be to extend the model of the vehicle with the dynamics of the ring. This way, the input of the system would no be τ_c anymore but the forces of each thruster and the angle of the ring. Doing so would also allow adding one more control layer on the behavior on the ring.

Of course, similar calculations can be realized in the second configuration presented on figure 5.10b where the rear thrusters are mounted on the ring and used to generate a surge force and a moment around any axis of the plane $(\boldsymbol{y_B}, \boldsymbol{z_B})$. When the two rear thrusters are mounted on the reconfigurable ring, the TCM becomes:

$$\boldsymbol{B}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -R & R \\ R\sin(\alpha_c) & -R\sin(\alpha_c) & 0 & 0 \\ -R\cos(\alpha_c) & R\cos(\alpha_c) & 0 & 0 \end{bmatrix} \tag{5.17}$$

In the third mode where the ring is used to compensate the roll angle of the vehicle, the ring angle is set to $\alpha_c = -\phi$ where ϕ is the current roll angle of the vehicle.

5.2.3 Control of PlaSMAR

Thanks to the allocation method detailed in the previous section, pretty much any controller presented in this work can be used on PlaSMAR. In the first configuration, the vehicle has five independent DOF: surge, sway, heave, roll and yaw. Because the three translations are independently actuated, no compensation mechanism is needed in position tracking tasks. Moreover, roll and yaw are actuated and pitch is usually naturally stabilized because of the mass distribution. The vehicle is therefore capable of keeping a neutral roll angle allowing rotations around the vertical axis of the inertial plane. Also, the association of roll and yaw allows the vehicle to roll to the side and get into an upright position allowing vertical diving. For the control, PID based controllers with a linearization mechanism or SMC can be used.

Same goes for the second configuration. The vehicle has 5 actuated DOF: surge,

heave, roll, pitch and yaw. Then, for a position tracking task like seabed scanning, roll or yaw can be used in association with heave or surge to compensate for sway. In such a case, the $\mathcal{H}$-matrix based controller or the SMC controller fro underactuated craft can be used. If roll is used for compensation and pitch is not actually part of the task, then the reconfiguration ring can be used to cancel out the roll angle and maintain the rear thrusters in the horizontal plane of the inertial frame. This way, the vehicle can be controlled in yaw at all times. (for any roll angle)

In the following, the vehicle in the first configuration is controlled using a very simple PID-based controller. The control effort vector is calculated in the inertial frame as:

$$\begin{aligned}\boldsymbol{\tau}_c = \tilde{\boldsymbol{M}}(\boldsymbol{\eta})\left(\ddot{\boldsymbol{\eta}}^* + \boldsymbol{K}_d\dot{\boldsymbol{e}}_\eta + \boldsymbol{K}_p\boldsymbol{e}_\eta + \boldsymbol{K}_i\int_0^t \boldsymbol{e}_\eta d\zeta\right)\\ + \tilde{\boldsymbol{C}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{D}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) \qquad (5.18a)\\ \tilde{\boldsymbol{M}}(\boldsymbol{\eta}) = \boldsymbol{M}\boldsymbol{J}(\boldsymbol{\eta})^{-1} \qquad (5.18b)\\ \tilde{\boldsymbol{C}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}) = \boldsymbol{M}\dot{\boldsymbol{J}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}})^{-1} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{J}(\boldsymbol{\eta})^{-1} \qquad (5.18c)\\ \tilde{\boldsymbol{D}}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}}) = \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{J}(\boldsymbol{\eta})^{-1} \qquad (5.18d)\end{aligned}$$

where $\boldsymbol{K}_d$, $\boldsymbol{K}_p$ and $\boldsymbol{K}_i$ are diagonal strictly definite gain matrices, $\ddot{\boldsymbol{\eta}}^*$ is the second order derivative of the desired state vector $\boldsymbol{\eta}^*$ and $\boldsymbol{e}_\eta = \boldsymbol{\eta}^* - \boldsymbol{\eta}$ is the error signal. The model matrices with the subscript $\tilde{\cdot}$ are expressed in the inertial frame using the kinematic relation $\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}$. Because the vehicle is not actuated in pitch, the fifth line of $\boldsymbol{\tau}_c$ is discarded.

5.3 Simulations

PlaSMAR is evaluated in simulation on several study cases. It is compared with the *RSM* AUV used before in this work and controlled using the $\mathcal{H}$-matrix based controller (see chapter 3 for more details) or a SMC inspired by [40]. In this SMC, the guidance action is created using sway error in the calculation of the yaw sliding surface. More details about this controller can be found in section 2.4.2.

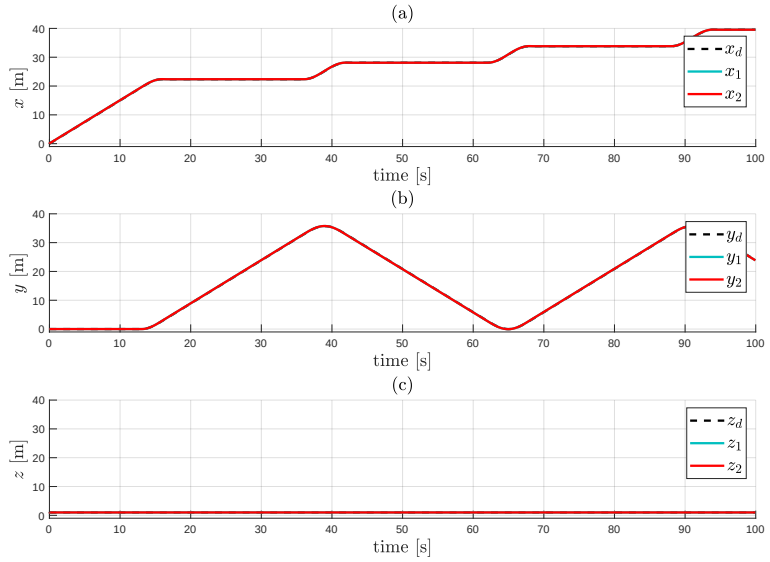


Figure 5.12 – Comparison of the position signals of the two vehicles. Blue: RSM, Red: PlaSMAR - a: $\boldsymbol{x_o}$ axis, b: $\boldsymbol{y_o}$ axis, c: $\boldsymbol{z_o}$ axis

5.3.1 Seabed scanning

In this first scenario, the first configuration of PlaSMAR is tested on the seabed scanning task. The vehicle is equipped with the front reconfigurable ring holding two parallel thrusters and two horizontal rear thrusters on the fixed ring. In this configuration, the vehicle is actuated in surge, sway, heave, roll and yaw and thus over-actuated with respect to the task which does not require roll. It is compared to the *RSM* vehicle which is ill-actuated on the task. It must use yaw to compensate for sway.

Figure 5.12 shows that both vehicles have perfect position tracking on the whole task. The main difference between the two vehicles appears in figure 5.13 where PlaSMAR (red line) appears to have a much smaller heading error than *RSM* (blue line). In facts, because the sway of PlaSMAR is directly actuated, the vehicle does not need any compensation mechanism and can truly control its yaw angle. This is not the case of *RSM* which uses yaw to compensate for sway. Numerical delay creates a slight heave and pitch oscillation on PlaSMAR.

To accentuate the difference between the two vehicles, they are evaluated on a second version of the seabed scanning task where the heading angle is constraint to $\psi^* = 0$.

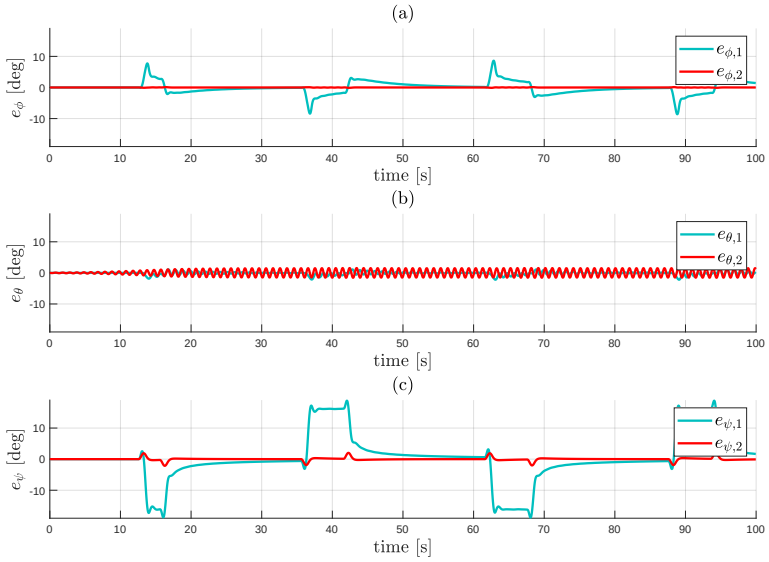
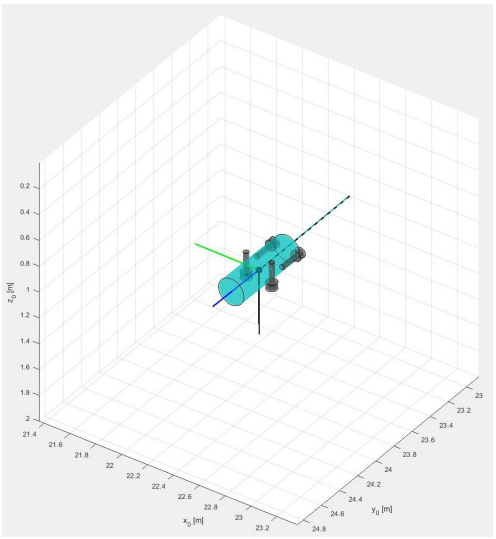


Figure 5.13 – Comparison of the orientation errors of the two vehicles. Blue: RSM, Red: PlaSMAR - a: roll, b: pitch, c: yaw

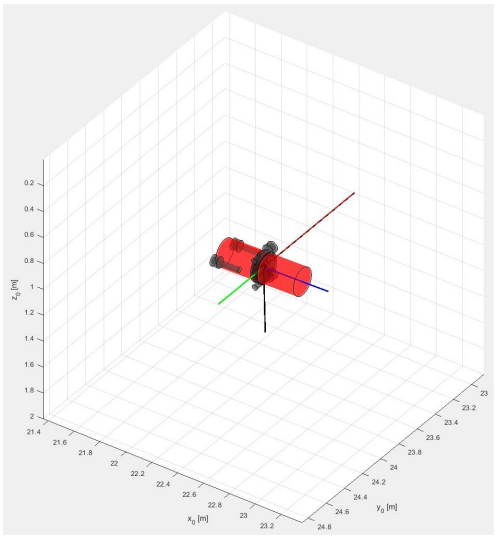
Figure 5.14 shows the two different behaviors of the vehicles. On figure 5.14a, *RSM* is tangent to the path aligned with the $\boldsymbol{y}_\mathbf{o}$ axis. Because yaw is used for compensation of sway, the heading constraint is neglected and the heading angle naturally stabilizes tangent to the velocity vector. On figure 5.14b on the other hand, PlaSMAR is aligned with the $\boldsymbol{x}_\mathbf{o}$ axis while traveling along the $\boldsymbol{y}_\mathbf{o}$ axis. The new DOF provided by the ring allows the vehicle to translate in sway while controlling its heading and maintaining constant depth.

5.3.2 Roll compensation

In this section, the reconfigurable ring of PlaSMAR is used for compensation of the roll of the vehicle. The second configuration shown on figure 5.10b is used where the two rear thrusters are mounted on a ring. The vehicle is therefore actuated in surge, heave, roll, pitch and yaw. In this example, it is evaluated on the seabed scanning. Because sway is not actuated, roll is used for compensation using the $\mathcal{H}_p$ matrix controller introduced in section 3.4.1. Section 3.4.1 shows that compensation mechanisms can



(a) *RSM* with SMC



(b) PlaSMAR with front reconfigruable ring (black)

Figure 5.14 – Simulation snapshots of the two vehicles on the seabed scanning with $\psi^* = 0$ - Blue axis: $\boldsymbol{x_B}$, Green axis: $\boldsymbol{y_B}$, Black axis: $\boldsymbol{z_B}$

disturb the other DOF of the systems. In particular in this case, using roll disturbs the heading control. With a non-zero roll angle, the capability of the vehicle to rotate aroudn the vertical axis of the inertial frame is diminished. At the extreme, when $\phi = \pi/2$ and $\theta = 0$, the $\boldsymbol{z_B}$ axis of the vehicle is in the horizontal plane of the inertial frame. Yaw actuators can no longer be used to control the heading of the craft.

To counteract this effect, the two rear thrusters are mounted on a rotating ring and maintained as fast as possible in the horizontal plane of the inertial frame. The angle command of the ring is at all time $\alpha_c = -\phi$.

This solution is compared with the *RSM* vehicle using the $\mathcal{H}_p$ based controller on the seabed scanning trajectory with the heading constraint $\psi^* = 0$.

Figure 5.15 shows the attitude of the two vehicles on the task. In the first turn around 15s, both vehicles roll to the side to generate the needed sway velocity at the tracking point but, because it stabilizes around $\phi = \pi/2$, the first vehicle (blue line) is not able to control its heading anymore. The two rear thrusters are in the vertical plane of $\mathcal{R}_0$. Then, the yaw control trying to maintain the heading of the vehicle destabilizes the system. On the other hand, the PlaSMAR vehicle stabilizes with $\phi = \pi/2$ and,thanks to the rotating ring, is still able to really control its heading.

Comparing figure 5.14b and figure 5.16 shows that both solutions are almost equiv-

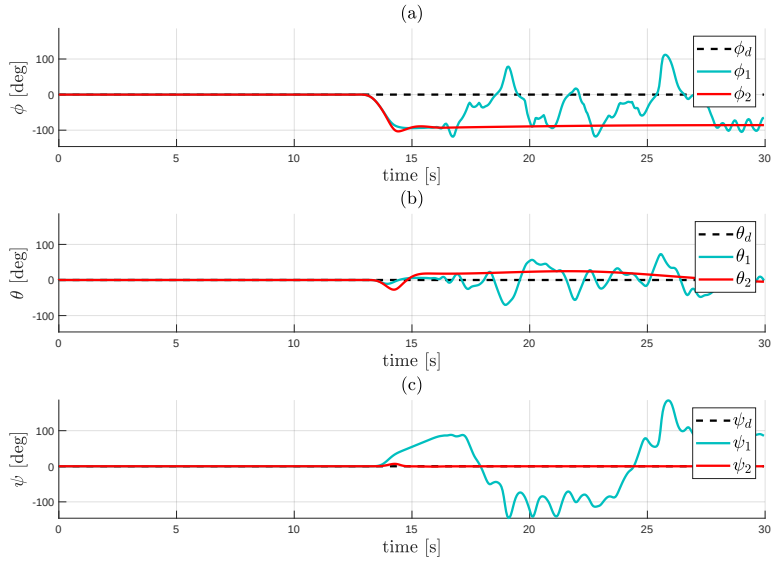


Figure 5.15 – Comparison of the attitude of the two vehicles. Blue: RSM, Red: PlaSMAR - a: roll, b: pitch, c: yaw

alent. The first vehicle keeps a neutral roll angle and directs the two front thrusters to generate sway while the second vehicle rolls to the side to be able to push along the $\boldsymbol{y}_\mathbf{o}$ axis while maintaining the two rear thrusters in the horizontal plane. This comparison also shows that one solution for this task with *RSM* or PlaSMAR would be to integrate the desired roll behavior of the vehicle directly in the task. Because both vehicles are actuated in roll and roll is not part of the task requirements, a roll trajectory mimicking the behavior of the vehicles obtained here could be calculated. The vessel would then track this roll angle and naturally align its thrusters to the the $\boldsymbol{y}_\mathbf{o}$ axis for instance to navigate along this axis without direct sway control. Such a solution is more about trajectory planning than it is about control.

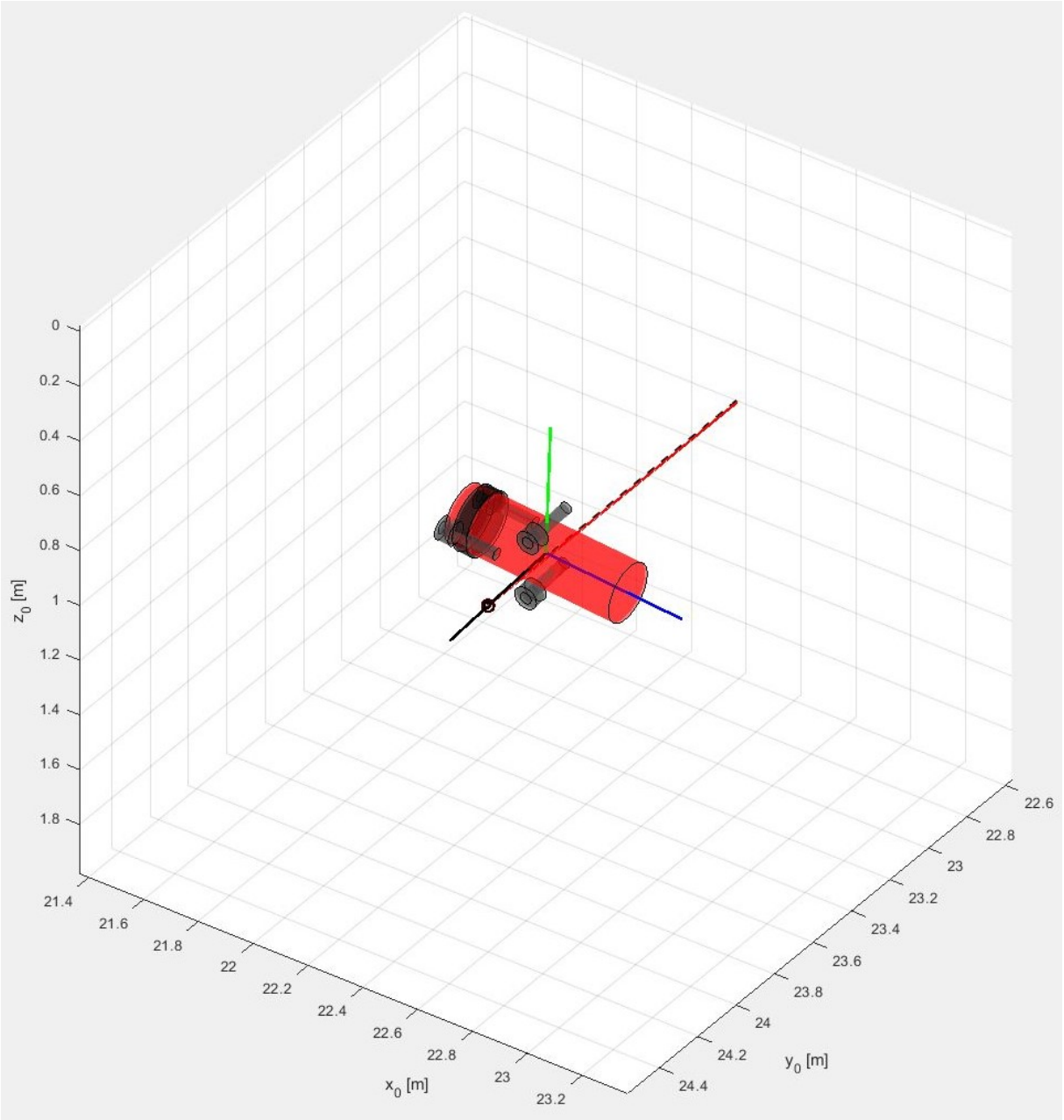


Figure 5.16 – Simulation snapshots of PlaSMAR with rear reconfigurable ring (black cylinder) on the seabed scanning with $\psi^* = 0$. - Blue axis: $\boldsymbol{x_B}$, Green axis: $\boldsymbol{y_B}$, Black axis: $\boldsymbol{z_B}$

5.4 Partial conclusions on vector thrust

A new concept of reconfigurable thrust has been introduced in this chapter: the reconfigurable ring. PlaSMAR is presented as a proof of concept of this new propulsive solution. The study of the TCM corresponding to the two configurations of PlaSMAR and simulation results exposed in this chapter show that the rotating ring allows adding a new DOF to the vehicle without creating new coupling. This new reconfiguration concept seems to be a better suited solution for smaller vehicles as it does not introduce any new disturbing effects.

The two configurations of PlaSMAR give new solutions to traditional control problem of underwater vehicles. The first configuration adds sway as an actuated DOF decoupled from the others. It facilitates position tracking as the vehicle does not require any compensation or guidance mechanism to track the three position components of the space.

The second configuration adds pitch. In this case, this new DOF can be used either as such, to generate a pitch moment on the vehicle, or to compensate the roll angle of the craft and maintain the rear thrusters in the horizontal plane of the inertial frame. These two solutions enhance the performances of the vehicle on the tasks presented here. It is worth noting that the rear rotating ring with two thrusters allows reproducing the surge, pitch and yaw actuation used on many usual underwater vehicles.

Further studies are required on this new technology. Notably, the actuation of the rotating ring is yet to be sorted out. Also, thrust allocation is not trivial on this system. More work on the propeller and reconfiguration dynamics is required to optimize the allocation.

CONCLUSION

This work follows on from the long line of work on the design and control of marine vehicles. Many types of different control laws and guidance principles for surface and underwater vehicles have been introduced and studied in details. All the solutions presented in the literature show equivalent theoretical performances. All of them ensure global or semi-global stability of the plant even in presence of external disturbances and noisy measurements.

Further work will be needed to properly compare all the different methods both mathematically and on experimental vehicles. Ranking the methods is very difficult as their performances will likely depend on the system itself. Other parameters will be to consider like computation time, effects of numerical sampling or the impact of poor-quality measurements.

Then, a new kinematic guidance principle based on the handy matrix $\mathcal{H}$ has been introduced. This new principle allows control of an underactuated or ill-actuated vehicle using kinematic couplings between a non actuated translation and an actuated rotation. The control is shown to be stable and robust to external disturbance and measurement noise. It will be easily extended to other types of systems thanks to the design rules and matrix calculation algorithm provided. These results also allow evaluating the capabilities of a given AUV and finding the tasks for which it is suited.

The controller has been successfully applied in both trajectory tracking and path following scenarios. Path following was performed on Remus100, a wide spread AUV representing the class of torpedo shape uqr-vessels. The $\mathcal{H}$-matrix based controller allows solving the path following problem in several manners. Using the $\mathcal{H}$-matrix based controller allows precisely choosing the behavior of the tracking point of the vehicle which is very valuable. In path following for instance, it allows finding simple PID-based solutions to the Lyapunov stability criterion. In future works, the controller could be used in other back-stepping applications to complete different types of tasks.

One of the interesting use cases of the path following example is the leader-follower scenario. In this case, the $\mathcal{H}$-matrix controller is used in the follower vehicle. The construction of the closed-loop system allows tracking the leader vehicle and adapting the

follower's behavior to ensure that both tasks are successful. Such applications could notably be interesting in pipeline surveillance where one vehicle, the leader, would be measuring or taking images of a submerged structure while the other, the follower, would go back and forth between the leader and the surface or a support ship.

Applying the controller to a physical vehicle would be valuable to assess its performances in conditions. Notably, it is very hard to anticipate the behavior of the control law with respect to numerical sampling, latency of the embedded calculator or unsteady measurement errors.

A flatness-based controller has also been developed using the more simple case of surface vessels. Flatness-based control appears to be a very strong contestant for the control of marine vehicles as it is shown to be very robust to model approximation and external disturbances. In this work, a flatness-based controller has been derived for surface vessels using the simplified model of the hovercraft. The controller has been developed under the hypothesis of a circular hull shape and even mass distribution but has been successfully applied to a generic surface vessel.

The flatness-based controller naturally includes a guidance principle. It allows calculating a surge force and a yaw moment out of two position signals. The flatness equations obtained in this example could also be used as a guidance principle with another type of controller.

For increased robustness and to counteract the potentially damaging behavior of flatness-based control away from the desired trajectory, it has been associated with an intelligent proportional derivative controller. The iPD increases the already impressive robustness of the flatness-based controller and effectively solves the problems caused by feedforward linearization when the vehicle is outside the acceptable bound around the desired trajectory. The combined controller has been shown to be stable and robust to external disturbances and noisy measurements. In this case again, the next step is to apply the controller on a physical system. Investigations on the application of the controller on a scale-model of the hovercraft have been started.

Also, some more calculation work is required to apply the controller to underwater vehicles. From the results obtained on the surface vessels, it is safe to assume that the controller will easily scale up to the underwater case. It seems notably well suited for uqr-vessels in position tracking applications.

Finally some new results on vectoring thrust applied to small scale AUVs have been presented. Notably, simulations have been conducted to show that the use of off-centered

vector thrusters is not suited for every type of vehicle. Although they are bounded, the coupling force effects generated by vector thrusters can jeopardize the tracking mission of smaller vehicles or of vehicles without natural robustness to force disturbances.

A new use case of vector thrusters on AUVs has also been exposed. While vector thrusters are usually used for moment control through dynamic couplings, simulation results show that they can also be used for force control. This new force control mode is shown to be especially useful for missions where moment control is naturally not suited. This is notably the case of autonomous missions like docking, hovering and station keeping missions.

The proof of concept for a new type of vectoring thrust system has also been proposed. The new system relies on pairs of thrusters mounted on a ring able to rotate around cylindrical sections of the hull of a vehicle. This rotating motion can be used to direct the force vector generated by the pair of thrusters while maintaining a stationary moment axis. When the thrusters are aligned with the axis of rotation of the ring, it allows directing the axis of the moment generated by the pair of thrusters while keeping the force vector aligned with the body of the vehicle. This second mode is shown to effectively unlock a new DOF for the vehicle as well as allowing compensation of roll disturbances.

On this matter, the next step is to design and build a second version of the test platform and investigate the possible solutions for actuation of the ring. Optimized magnetic couplings will be investigated as they would allow piloting the rotation of the ring from the inside of the hull therefore avoiding the need of holes in the hull.

Some more work will be conducted on the controller of this vehicle. A controller taking the dynamics of the ring into account would be interesting and might appear to be necessary. The model of the vehicle would have to be extended to take the behavior of the ring into account and include the ring angle as a new control input. In the same way, further investigations on optimizing the allocation procedure are required for increased performances.

Overall, this work proposes several solutions to increasing the maneuverability of underactuated marine craft through control and new mechanical designs.

Appendices

APPENDIX A

INTEGRAL SLIDING MODE CONTROL

This appendix demonstrates that steady state can appear when using a sliding mode controller on a system subjected to external unknown disturbances or model approximations. Then, *Integral Sliding Mode* is presented as a solution against this steady state error. There are not many references mentioning integral SMC in the literature but some hints can be found in [108, 150]. Note that *super-twisting* SMC as described for instance in [151] has an integral behavior and cancels steady state error.

A demonstration based on the study of the equilibrium of the closed loop system is provided. This phenomena appears to be rather absent from the literature and notably in the context of marine craft although it can cause problems in many real-life applications. In this example, the demonstration is performed on a fully-actuated system but similar effects can be shown on underactuated vehicles.

A.1 Apparition of steady state error

To demonstrate the apparition of steady state error, the model of a marine craft expressed in the inertial frame and in presence of an unknown external disturbance is introduced.

$$\boldsymbol{\tau} = \bar{\boldsymbol{M}}(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \bar{\boldsymbol{C}}(\boldsymbol{\eta},\boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \bar{\boldsymbol{D}}(\boldsymbol{\eta},\boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) + \Delta \tag{A.1}$$

Matrices $\bar{\boldsymbol{M}}$, $\bar{\boldsymbol{C}}$ and $\bar{\boldsymbol{D}}$ are the model matrices modified to express the model in the inertial frame. They contain the transformation based on the $\boldsymbol{J}(\boldsymbol{\eta})$ matrix. Vector Δ marks the unknown external disturbance effect.

Following the reasoning introduced in the examples of section 2.4, the sliding surface for this second order system is defined with respect to $\boldsymbol{\eta}$ as:

$$\boldsymbol{\sigma} = (\frac{d}{dt} + \boldsymbol{\lambda})^{(2-1)}\boldsymbol{e}_{\eta} = \dot{\boldsymbol{e}}_{\eta} + \boldsymbol{\lambda}\boldsymbol{e}_{\eta} \tag{A.2}$$

A simple sliding condition is used in this example as the super-twisting and terminal sliding mode conditions exposed in section 2.4 do not make much difference on this issue. To avoid chattering, a saturation function is used instead of the intuitive signum function. Note that the signum function does not solve the steady state error. The sliding condition is then:

$$\dot{\boldsymbol{\sigma}} = -\boldsymbol{K}\text{sat}(\boldsymbol{\sigma}) \tag{A.3}$$

Using the method described in section 2.4, the control is established from the sliding condifiton and the model equation:

$$\boldsymbol{\tau}_c = \bar{\boldsymbol{M}}(\boldsymbol{\eta})(\ddot{\boldsymbol{\eta}}^* + \boldsymbol{\lambda}\dot{\boldsymbol{e}}_\eta + \boldsymbol{K}\text{sat}(\boldsymbol{\sigma})) + \bar{\boldsymbol{C}}(\boldsymbol{\eta},\boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \bar{\boldsymbol{D}}(\boldsymbol{\eta},\boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{A.4}$$

Of course, because Δ is an unknown disturbance term, it cannot be used in the control calculations. The closed-loop system is therefore:

$$\ddot{\boldsymbol{e}}_\eta + \boldsymbol{\lambda}\dot{\boldsymbol{e}}_\eta + \boldsymbol{K}\text{sat}(\boldsymbol{\sigma}) = \bar{\boldsymbol{M}}^{-1}\Delta \tag{A.5}$$

A quick way to demonstrate the appearance of a steady state error in to study the values taken by the error vector $\boldsymbol{e}_\eta$ when an equilibrium $\ddot{\boldsymbol{e}}_\eta = \dot{\boldsymbol{e}}_\eta = 0$ is reached. To do so, the system is considered in the neighborhood of the sliding surface meaning $\text{sat}(\boldsymbol{\sigma}) = \boldsymbol{\sigma}$. In such conditions, the closed-loop system becomes :

$$\boldsymbol{K}\boldsymbol{\lambda}\boldsymbol{e}_\eta = \bar{\boldsymbol{M}}^{-1}\Delta \tag{A.6}$$

In this conditions, it appears clearly that a steady state error depending on the disturbance magnitude appears at the error equilibrium. This steady state can be made as small as necessary tuning the gain parameters $\boldsymbol{K}$ and $\boldsymbol{\lambda}$ but cannot be canceled. In facts, inside the boundary layer, the SMC behaves similarly to a Proportional Derivative controller hence the steady state error.

A.2 Integral Sliding Mode

When working with Proportional and Proportional Derivative controllers, the solution to cancel steady state errors is to add an integral term. The same idea is used in *Integral* Sliding Mode Control (ISMC). Several definitions of the ISMC can be found

but it is rarely seen in the context of marine vehicles. The ISMC used in this example consists in defining the sliding surface with respect to the integral of the position and orientation error instead of the error. Same reasoning can be seen in some underactuated examples of section 2.4 where the integral of the speed signal is used to take constant position offsets into account. The system (A.3) is then considered of the third order with respect to the integral of $\boldsymbol{\eta}$. The sliding surface is therefore defined as:

$$\boldsymbol{\sigma} = (\frac{d}{dt} + \boldsymbol{\lambda})^{(3-1)} \int_0^t \boldsymbol{e}_{\boldsymbol{\eta}}(\zeta)d\zeta = \dot{\boldsymbol{e}}_{\boldsymbol{\eta}} + 2\boldsymbol{\lambda}\boldsymbol{e}_{\boldsymbol{\eta}} + \boldsymbol{\lambda}^2 \int_0^t \boldsymbol{e}_{\boldsymbol{\eta}}(\zeta)d\zeta \tag{A.7}$$

The same sliding condition is used as in equation (A.3):

$$\dot{\boldsymbol{\sigma}} = -\boldsymbol{K}\text{sat}(\boldsymbol{\sigma}) = \ddot{\boldsymbol{e}}_{\boldsymbol{\eta}} + 2\boldsymbol{\lambda}\dot{\boldsymbol{e}}_{\boldsymbol{\eta}} + \boldsymbol{\lambda}^2\boldsymbol{e}_{\boldsymbol{\eta}} \tag{A.8}$$

In these conditions and following the same logic as before, the controller is then:

$$\boldsymbol{\tau}_c = \bar{\boldsymbol{M}}(\boldsymbol{\eta})(\ddot{\boldsymbol{\eta}}^* + 2\boldsymbol{\lambda}\dot{\boldsymbol{e}}_{\boldsymbol{\eta}} + \boldsymbol{\lambda}^2\boldsymbol{e}_{\boldsymbol{\eta}} + \boldsymbol{K}\text{sat}(\boldsymbol{\sigma})) + \bar{\boldsymbol{C}}(\boldsymbol{\eta}, \boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \bar{\boldsymbol{D}}(\boldsymbol{\eta}, \boldsymbol{\nu})\dot{\boldsymbol{\eta}} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{A.9}$$

In this case, the closed-loop system becomes:

$$\ddot{\boldsymbol{e}}_{\boldsymbol{\eta}} + 2\boldsymbol{\lambda}\dot{\boldsymbol{e}}_{\boldsymbol{\eta}} + \boldsymbol{\lambda}^2\boldsymbol{e}_{\boldsymbol{\eta}} + \boldsymbol{K}\text{sat}(\boldsymbol{\sigma}) = \bar{\boldsymbol{M}}^{-1}\Delta \tag{A.10}$$

As before, the equilibrium in the neighborhood of the sliding surface is studied: $\text{sat}(\boldsymbol{\sigma}) = \boldsymbol{\sigma}$, $\ddot{\boldsymbol{e}}_{\boldsymbol{\eta}} = \dot{\boldsymbol{e}}_{\boldsymbol{\eta}} = 0$. The closed-loop system is rewritten as:

$$\boldsymbol{e}_{\boldsymbol{\eta}} = (\boldsymbol{\lambda} + 2\boldsymbol{K}\boldsymbol{\lambda})^{-1} \left(\bar{\boldsymbol{M}}^{-1}\Delta - \boldsymbol{K}\boldsymbol{\lambda}^2 \int_0^t \boldsymbol{e}_{\boldsymbol{\eta}}(\zeta)d\zeta \right) \tag{A.11}$$

Equation (A.11) shows that the state error $\boldsymbol{e}_{\boldsymbol{\eta}}$ can be canceled at the equilibrium thanks to the integral term. The ISMC formulation therefore allows to cancel steady state errors due to external unknown disturbances.

APPENDIX B

ALTERNATIVE REPRESENTATION OF THE GENERALIZED STATE SYSTEM

This appendix gives an alternative representation to the generalized state system described in section 4.3.3. The system is recalled here:

$$\dot{x} = u\cos\psi - v\sin\psi \tag{B.1a}$$
$$\dot{y} = u\sin\psi + v\cos\psi \tag{B.1b}$$
$$\dot{u} = \tilde{\tau}_u + v\dot{\psi} + \beta u \tag{B.1c}$$
$$\dot{v} = -u\dot{\psi} + \beta v \tag{B.1d}$$

The inputs of system (B.1) are the surge force $\tilde{\tau}_u$ and the yaw angle ψ whom first order derivative appears in equation (B.1c) and equation (B.1d).

Following the method exposed in [37, 38], a generalized state transformation is calculated to cancel the dependency in $\dot{\psi}$. The infinite prolongation [184] is introduced for system (B.1):

$$\begin{aligned}\frac{d}{dt} = (u\cos\psi - v\sin\psi)\frac{\partial}{\partial x} + (u\sin\psi + v\cos\psi)\frac{\partial}{\partial y} \\ + (\tilde{\tau}_u + v\dot{\psi} + \beta u)\frac{\partial}{\partial u} + (-u\dot{\psi} + \beta v)\frac{\partial}{\partial v} \\ + \sum_{k=0}^{\infty}\tilde{\tau}_u^{(k+1)}\frac{\partial}{\partial\tilde{\tau}_u^{(k)}} + \sum_{k=0}^{\infty}\psi^{(k+1)}\frac{\partial}{\partial\psi^{(k)}}\end{aligned} \tag{B.2}$$

The new state $\boldsymbol{\zeta} = [\zeta_1\ \zeta_2\ \zeta_3\ \zeta_4]^T$ must satisfy the following condition expressed in terms of Lie bracket:

$$\left[\frac{d}{dt}, \frac{\partial}{\partial\dot{\psi}}\right](\boldsymbol{\zeta}) = 0 \tag{B.3}$$

This partial differential equation ensures that the first time derivative of the new state $\boldsymbol{\zeta}$ does not depend on $\dot{\psi}$. The Lie bracket thus reads :

$$\left[\frac{d}{dt}, \frac{\partial}{\partial \dot{\psi}}\right] = v\frac{\partial}{\partial u} - u\frac{\partial}{\partial v} + \frac{\partial}{\partial \psi} \tag{B.4}$$

The four components of the new state $\boldsymbol{\zeta}$ are chosen as independent functions of the state variables. The natural choice for the first two components ζ_1 and ζ_2 is x and y respectively. The position coordinates obviously respect condition (B.3):

$$(v\frac{\partial}{\partial u} - u\frac{\partial}{\partial v} + \frac{\partial}{\partial \psi})(x) = 0 \tag{B.5a}$$

$$(v\frac{\partial}{\partial u} - u\frac{\partial}{\partial v} + \frac{\partial}{\partial \psi})(y) = 0 \tag{B.5b}$$

Then, two more independent functions must be found for ζ_3 and ζ_4. A multitude of solutions exist for this problem so two fairly simple solutions are selected:

$$\zeta_3 = \frac{1}{2}(u^2 + v^2) \tag{B.6a}$$

$$\zeta_4 = \psi + \text{ atan}\left(\frac{v}{u}\right) \tag{B.6b}$$

The two solutions ζ_3 and ζ_4 of equation (B.6) are shown to be suitable with respect to condition (B.3):

$$\left[\frac{d}{dt}, \frac{\partial}{\partial \dot{\psi}}\right](\zeta_3) = vu - uv = 0 \tag{B.7a}$$

$$\left[\frac{d}{dt}, \frac{\partial}{\partial \dot{\psi}}\right](\zeta_4) = -\frac{v^2}{u^2 + v^2} - \frac{u^2}{u^2 + v^2} + 1 = 0 \tag{B.7b}$$

The state $\boldsymbol{\zeta}$ is then given as:

$$\zeta_1 = x \tag{B.8a}$$

$$\zeta_2 = y \tag{B.8b}$$

$$\zeta_3 = \frac{1}{2}(u^2 + v^2) \tag{B.8c}$$

$$\zeta_4 = \psi + \text{ atan}\left(\frac{v}{u}\right) \tag{B.8d}$$

and the associated new state space representation is:

$$\dot{\zeta}_1 = \dot{x} \tag{B.9a}$$

$$\dot{\zeta}_2 = \dot{y} \tag{B.9b}$$

$$\dot{\zeta}_3 = (\dot{\zeta}_1 \cos\psi + \dot{\zeta}_2 \sin\psi)\tau_u + 2\beta\zeta_3 \tag{B.9c}$$

$$\dot{\zeta}_4 = -\frac{\dot{\zeta}_1 \sin\psi + \dot{\zeta}_2 \cos\psi}{2\zeta_3}\tau_u \tag{B.9d}$$

To conclude, the new state space representation is indeed standard, the state transformation canceled all dependency in the input derivative $\dot{\psi}$. The standard representation could then be used to perform the controller calculations.

Note that, in this specific case, the Brunovsky state transformation detailled in section 4.3.3.2 also cancels the dependencies in $\dot{\psi}$ because of the symmetries in the plant but this is not a general result.

APPENDIX C

CONTROL OF UNDERACTUATED SYSTEMS WITH COUPLED ACTUATORS

This appendix presents a side effect of the control of some underactuated systems independently from the control method used. In these examples, a feedback linearizing controller is used as it is the most frequently used in this work. This phenomena notably appears when coupled actuators are used to control underactuated systems. In such cases, one of the two coupled effects generated by an actuator (most often one force and one coupled moment) is chosen for the control of the system. The other one can create non negligible disturbance on the uncontrolled DOF or even on the whole system.

Most examples of such systems rely on the natural stability of the uncontrolled DOF (for instance autonomous boats). Yet, the passive stability of the free DOF may not be robust enough to counteract the effect of the coupled actuator. It might notably be the case for small scale vehicle equipped with vector thrusters or vehicles with simple hull shapes (cylinder, box shape).

Consider the following control system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{C.1}$$

where $\boldsymbol{x} \in \mathbb{R}^3$ is the state vector, $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^3$ is a vector field, $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^{3,3}$ is a known matrix and $\boldsymbol{u} \in \mathbb{R}^3$ is the input vector.

Considering that $\boldsymbol{g}(\boldsymbol{x})$ is invertible, an intuitive model-based feedback linearizing controller can be designed as:

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x})^{-1}\left(\dot{\boldsymbol{x}}_d + \boldsymbol{\lambda}(\boldsymbol{e}_x) - \boldsymbol{f}(\boldsymbol{x})\right) \tag{C.2}$$

where $\boldsymbol{\lambda}(\boldsymbol{e}_x)$ is a control law based on the tracking error $\boldsymbol{e}_x$ and $\dot{\boldsymbol{x}}_d$ is the desired velocity used as a feedforward term. Plugging the controller (C.2) into the model equation (C.1)

gives the closed-loop system:

$$\dot{\boldsymbol{e_x}} = \boldsymbol{\lambda}(\boldsymbol{e_x}) \tag{C.3}$$

the non-linear terms of the model are canceled using feedback linearization and choosing function $\boldsymbol{\lambda}$ properly, stability of the equilibrium $\boldsymbol{e_x} = 0$ can be ensured for all three DOF of the space.

Now, let us consider the same system but in the underactuated context. In this second example, the second component of the control vector $\boldsymbol{u}$ is always 0 because of underactuation:

$$\boldsymbol{u} = \begin{bmatrix} u_1 & 0 & u_3 \end{bmatrix}^T \tag{C.4}$$

The controller is built again using the intuitive feedback linearizing solution equation (C.2). This time though, because of underactuation, the three DOF of the system behave differently. Let us define the vectors and matrices of the system as:

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T \tag{C.5a}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) & f_2(\boldsymbol{x}) & f_3(\boldsymbol{x}) \end{bmatrix}^T \tag{C.5b}$$

$$\boldsymbol{\lambda}(\boldsymbol{e_x}) = \begin{bmatrix} \lambda_1(\boldsymbol{e_x}) & \lambda_2(\boldsymbol{e_x}) & \lambda_3(\boldsymbol{e_x}) \end{bmatrix}^T \tag{C.5c}$$

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} g_1(\boldsymbol{x}) & 0 & 0 \\ 0 & g_2(\boldsymbol{x}) & 0 \\ 0 & 0 & g_3(\boldsymbol{x}) \end{bmatrix} \tag{C.5d}$$

Note that all three components of $\boldsymbol{f}$ can depend on the complete state vector $\boldsymbol{x}$. In this first case, $\boldsymbol{g}(\boldsymbol{x}))$ has been chosen diagonal. All three DOF are decoupled.

Then, applying the controller (C.2) with the hypothesis (C.4) on the system (C.1)

considering (C.5) gives:

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ 0 \\ u_3 \end{bmatrix} = \begin{bmatrix} g_1^{-1}(\boldsymbol{x}) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & g_3^{-1}(\boldsymbol{x}) \end{bmatrix} \left(\begin{bmatrix} \dot{x}_{1,d} \\ 0 \\ \dot{x}_{3,d} \end{bmatrix} \begin{bmatrix} \lambda_1(\boldsymbol{e_x}) \\ 0 \\ \lambda_3(\boldsymbol{e_x}) \end{bmatrix} - \begin{bmatrix} f_1(\boldsymbol{x}) \\ 0 \\ f_3(\boldsymbol{x}) \end{bmatrix} \right) \tag{C.6a}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} g_1(\boldsymbol{x}) & 0 & 0 \\ 0 & g_2(\boldsymbol{x}) & 0 \\ 0 & 0 & g_3(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} u_1 \\ 0 \\ u_3 \end{bmatrix} \tag{C.6b}$$

Note that the second line of $\boldsymbol{u}$ has been artificially set to zero to respect the constraint of underactuation. The closed-loop system becomes:

$$\dot{x}_1 = \dot{x}_{1,d} + \lambda_1(\boldsymbol{e_x}) \tag{C.7a}$$

$$\dot{x}_2 = f_2(\boldsymbol{x}) \tag{C.7b}$$

$$\dot{x}_3 = \dot{x}_{3,d} + \lambda_3(\boldsymbol{e_x}) \tag{C.7c}$$

Equation (C.7) shows that the first and third components of $\boldsymbol{x}$ can indeed be driven towards the desired value using the λ_1 and λ_3 functions but that the second component x_2 is left uncontrolled. Moreover, since the function $f_2(\boldsymbol{x})$ can depend on the complete state $\boldsymbol{x}$ (natural coupling terms), the behavior of x_2 is hardly predictable, could be unstable and could destabilize the rest of the system.

Furthermore, the matrix $\boldsymbol{g}(\boldsymbol{x})$ could contain off-diagonal coupling terms:

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} g_1(\boldsymbol{x}) & 0 & 0 \\ 0 & g_{2,2}(\boldsymbol{x}) & g_{2,3}(\boldsymbol{x}) \\ 0 & 0 & g_3(\boldsymbol{x}) \end{bmatrix} \tag{C.8}$$

In such a case, the system becomes:

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ 0 \\ u_3 \end{bmatrix} = \begin{bmatrix} g_1^{-1}(\boldsymbol{x}) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & g_3^{-1}(\boldsymbol{x}) \end{bmatrix} \left(\begin{bmatrix} \dot{x}_{1,d} \\ 0 \\ \dot{x}_{3,d} \end{bmatrix} \begin{bmatrix} \lambda_1(\boldsymbol{e_x}) \\ \lambda_2(\boldsymbol{e_x}) \\ \lambda_3(\boldsymbol{e_x}) \end{bmatrix} - \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \end{bmatrix} \right) \tag{C.9a}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ f_3(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} g_1(\boldsymbol{x}) & 0 & 0 \\ 0 & g_{2,2}(\boldsymbol{x}) & g_{2,3}(\boldsymbol{x}) \\ 0 & 0 & g_3(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} u_1 \\ 0 \\ u_3 \end{bmatrix} \tag{C.9b}$$

thus the closed-loop system becomes:

$$\dot{x}_1 = \dot{x}_{1,d} + \lambda_1(\boldsymbol{e_x}) \tag{C.10a}$$

$$\dot{x}_2 = f_2(\boldsymbol{x}) + g_{2,3}(\boldsymbol{x}) g_3(\boldsymbol{x})^{-1} \left(\dot{x}_{3,d} + \lambda_3(\boldsymbol{e_x}) - f_3(\boldsymbol{x}) \right) \tag{C.10b}$$

$$\dot{x}_3 = \dot{x}_{3,d} + \lambda_3(\boldsymbol{e_x}) \tag{C.10c}$$

Equation (C.10) shows that the third control component u_3 could be used to control the second DOF of the system x_2 through the coupling term $g_{2,3}$ but at the cost of the control over the thrid DOF x_3 which would be left uncontrolled. If x_3 is controlled then u_3 creates disturbance on x_2 that could make it even more unstable.

In many systems, the stability of the non-actuated DOF is assumed to be robust enough to counteract disturbances created by the actuators. It is notably the case for most boats where the hull shape, mass distribution and hydrodynamic effects damp any sway motion and favor surge and yaw displacements. Therefore, even if the vehicle is actuated in surge and yaw only, the natural sway stability allows controlling either the yaw angle of the vehicle or the two position coordinates without risking unstability. Nonetheless, other systems like the small cylindrical hull underwater vehicle used in some examples of this work (see chapter 1) are not guaranteed to be stable enough. Coupled actuators may create instability of the uncontrolled DOF. The physical bounds of the actuators ensure that the disturbances created by coupled actuators are bounded too.

APPENDIX D

MODEL COEFFICIENTS OF THE VEHICLES

This appendix gives the numerical values of th model coefficients of the robots used in this work: *RSM* , Remus100 and ODIN.

RSM			
Item	**Value**	**Item**	**Value**
m	$18\,\mathrm{kg}$	I_x	0.096
I_y	0.611	I_z	0.608
$\boldsymbol{P_G}$	$[0, 0, 0.012]^\top$	$\boldsymbol{P_B}$	$[0, 0, 0]^\top$
$X_{\dot{u}}$	-1.88	$Y_{\dot{v}}$	-18.81
$Z_{\dot{w}}$	-18.81	$K_{\dot{p}}$	0
$M_{\dot{q}}$	-0.56	$N_{\dot{r}}$	-0.56
X_u	-12.85	Y_v	-71.85
Z_w	-71.85	K_p	-0.38
M_q	-3.88	N_r	-3.88

Table D.1 – Simulation parameters - *RSM*

Remus100			
Item	**Value**	**Item**	**Value**
m	$31\,\mathrm{kg}$	I_x	0.17
I_y	6.70	I_z	6.70
$\boldsymbol{P_G}$	$[0, 0, 0.02]^\top$	$\boldsymbol{P_B}$	$[0, 0, 0]^\top$
$X_{\dot{u}}$	-0.83	$Y_{\dot{v}}$	-29.43
$Z_{\dot{w}}$	-29.43	$K_{\dot{p}}$	-0.033
$M_{\dot{q}}$	-3.42	$N_{\dot{r}}$	-3.42
X_u	-1.59	Y_v	-3.02
Z_w	-3.02	K_p	-0.58
M_q	-10.78	N_r	-1.49
X_{uu}	-12.85	Y_{vv}	-1.91
Z_{ww}	-1.91	K_{pp}	-0.38
M_{qq}	-1.96	N_{rr}	-1.96

Table D.2 – Simulation parameters - Remus100

ODIN			
Item	**Value**	**Item**	**Value**
m	$84\,\mathrm{kg}$	I_x	3.79
I_y	2.52	I_z	2.52
$\boldsymbol{P_G}$	$[0,0,0]^\top$	$\boldsymbol{P_B}$	$[0,0,0]^\top$
$X_{\dot{u}}$	-62.50	$Y_{\dot{v}}$	-62.50
$Z_{\dot{w}}$	-62.501	$K_{\dot{p}}$	0
$M_{\dot{q}}$	-30	$N_{\dot{r}}$	-30
X_u	-48	Y_v	-48
Z_w	-48	K_p	-80
M_q	-80	N_r	-80

Table D.3 – Simulation parameters - ODIN

BIBLIOGRAPHY

[1] A.P. Aguiar and J.P. Hespanha. « Position tracking of underactuated vehicles ». en. In: *Proceedings of the 2003 American Control Conference, 2003.* Vol. 3. Denver, CO, USA: IEEE, 2003, pp. 1988–1993. DOI: 10.1109/ACC.2003.1243366.

[2] M. A. Aizerman and F. R. Gantmacher. « On some features of switchings in nonlinear control with a piecewise smooth response of the nonlinear element ». In: *Automatika i Telemekhanika* 18.11 (1957).

[3] F. Alonge, F. D'Ippolito, and F.M. Raimondi. « Trajectory Tracking of Underactuated Underwater Vehicles ». In: *40th IEEE Conference on Decision and Control (CDC)*. 2001.

[4] M. S. Anandan and P. S. Lal Priya. « Super Twisting Sliding Mode Controller for a Diving Autopilot ». In: *2nd International Conference on Power, Control and Computing Technologies (ICPC2T)*. Raipur, India, 2022, pp. 1–6.

[5] David Angeli and Eduardo D. Sontag. « Forward completeness, unboundedness observability, and their Lyapunov characterizations ». en. In: *Systems & Control Letters* 38.4-5 (Dec. 1999), pp. 209–217. DOI: 10.1016/S0167-6911(99)00055-9.

[6] G. Antonelli, F. Caccavale, S. Chiaverini, and G. Fusco. « A novel adaptive control law for underwater vehicles ». In: *IEEE Transactions on Control Systems Technology* 11.2 (2003), pp. 221–232.

[7] Gianluca Antonelli. « On the Use of Adaptive/Integral Actions for Six-Degrees-of-Freedom Control of Autonomous Underwater Vehicles ». In: *IEEE Journal of Oceanic Engineering* 32.2 (Apr. 2007). Conference Name: IEEE Journal of Oceanic Engineering, pp. 300–312. DOI: 10.1109/JOE.2007.893685.

[8] Gianluca Antonelli. *Underwater Robots*. 4th. Cham: Springer International Publishing, 2018.

[9] Gianluca Antonelli, Fabrizio Caccavale, Stefano Chiaverini, and Giuseppe Fusco. « On the use of integral control actions for autonomous underwater vehicles ». In: *European Control Conference (ECC)*. Porto, 2001, pp. 1186–1191.

[10] Hashem Ashrafiuon and Kenneth R. Muske. « Sliding mode tracking control of surface vessels ». In: *IEEE American Control Conference (ACC)*. Seattle, WA, 2008, pp. 556–561.

[11] Jacson Mo Barth, Jean-Philippe Condomines, Murat Bronz, Jean-Marc Moschetta, Cédric Join, and Michel Fliess. « Model-free control algorithms for micro air vehicles with transitioning flight capabilities ». en. In: *International Journal of Micro Air Vehicles* 12 (Jan. 2020), p. 175682932091426. DOI: 10.1177/1756829320914264.

[12] Charalampos P. Bechlioulis, George C. Karras, Shahab Heshmati-Alamdari, and Kostas J. Kyriakopoulos. « Trajectory Tracking With Prescribed Performance for Underactuated Underwater Vehicles Under Model Uncertainties and External Disturbances ». In: *IEEE Transactions on Control Systems Technology* 25.2 (Mar. 2017). Conference Name: IEEE Transactions on Control Systems Technology, pp. 429–440. DOI: 10.1109/TCST.2016.2555247.

[13] M. K. Bennani and Pierre Rouchon. « Robust stabilization of flat and chained systems ». In: *3rd European Control Conference (ECC)*. Roma, Italy, 1995, pp. 1781–1786.

[14] Svein P. Berge, Kohei Ohtsu, and Thor I. Fossen. « Nonlinear control of ships minimizing the position tracking errors ». en. In: *MIC* 20.3 (1999), pp. 177–187. DOI: 10.4173/mic.1999.3.3.

[15] Zafer Bingul and Kursad Gul. « Intelligent-PID with PD Feedforward Trajectory Tracking Control of an Autonomous Underwater Vehicle ». en. In: *Machines* 11.2 (Feb. 2023), p. 300. DOI: 10.3390/machines11020300.

[16] Even Borhaug, A. Pavlov, and Kristin Y. Pettersen. « Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents ». In: *47th IEEE Conference on Decision and Control (CDC)*. Cancun, Mexico, 2008, pp. 4984–4991.

[17] F. Boyer, M. Porez, and W. Khalil. « Macro-continuous computed torque algorithm for a three-dimensional eel-like robot ». In: *IEEE Trans. Robot.* 22.4 (Aug. 2006), pp. 763–775. DOI: 10.1109/TRO.2006.875492.

[18] Frédéric Boyer, Damien Chablat, Philippe Lemoine, and Philippe Wenger. « The eel-like robot ». In: (2009). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.0908.4464.

[19] M. Breivik and T.I. Fossen. « A unified control concept for autonomous underwater vehicles ». In: *IEEE American Control Conference (ACC)*. 2006.

[20] M. Breivik and T.I. Fossen. « Principles of Guidance-Based Path Following in 2D and 3D ». In: *IEEE Conference on Decision and Control (CDC)*. Seville, Spain, 2005, pp. 627–634.

[21] Morten Breivik and Thor I. Fossen. « Guidance laws for autonomous underwater vehicles ». In: *Underwater vehicles* 4 (2009). Publisher: InTech, pp. 51–76.

[22] Brockett. « Feedback invariants for non-linear systems ». In: *7th IFAC World Congress*. Helsinki, 1978, pp. 1115–1120.

[23] Pavol Brunovský. « A classification of linear controllable systems ». eng. In: *Kybernetika* 06.3 (1970), (173)–188.

[24] Walter Caharija, Kristin Y. Pettersen, Jan Tommy Gravdahl, and Even Borhaug. « Integral LOS guidance for horizontal path following of underactuated autonomous underwater vehicles in the presence of vertical ocean currents ». In: *IEEE American Control Conference (ACC)*. Montreal, QC, 2012, pp. 5427–5434.

[25] Walter Caharija, Kristin Y. Pettersen, Jan Tommy Gravdahl, and Even Borhaug. « Path following of underactuated autonomous underwater vehicles in the presence of ocean currents ». In: *51st IEEE Conference on Decision and Control (CDC)*. Maui, HI, USA, 2012, pp. 528–535.

[26] O. Calvo, A. Rozenfeld, A. Souza, F. Valenciaga, P.F. Puleston, and G. Acosta. « Experimental results on smooth path tracking with application to pipe surveying on inexpensive AUV ». In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*. Nice, 2008, pp. 3647–3653.

[27] R. Charlet, Jean Lévine, and Riccardo Marino. « On dynamic feedback linearization ». In: *Systems & Control Letters* 13 (1989), pp. 143–151.

[28] A. Chelouah and M. Petitot. « Finitely discretizable nonlinear systems: concepts and definitions ». In: *34th IEEE Conference on Decision and Control (CDC)*. Vol. 1. New Orleans, LA, USA, 1995, pp. 19–24.

[29] Abdelkader Chelouah, Emmanuel Delaleau, Philippe Martin, and Pierre Rouchon. « Differential flatness and control of induction motors ». In: *Proceedings of the Symposium on Control, Optimization and Supervision; Computational Engineering in Systems Applications IMACS Multiconference*. Ed. by P. Borne, M. Staroswiecki, J. P. Cassar, and S. El Khattabi. (Invited paper). Lille, France, 1996, pp. 80–85.

[30] Olivier Chocron and Emmanuel Delaleau. « Task-based design of AUVs propulsion systems including fixed or vector thrusters ». In: *45th Annual Conference of the IEEE Industrial Electronics Society (IECON)*. Vol. 1. 2019, pp. 5292–5298.

[31] Olivier Chocron, Emanuel P. Vega, and Mohamed Benbouzid. « Dynamic reconfiguration of autonomous underwater vehicles propulsion system using genetic optimization ». en. In: *Ocean Engineering* 156 (2018), pp. 564–579.

[32] Daniel Claude. « Everything you always wanted to know about linearization (but you were afraid to ask) ». In: *Algebraic and Geometric Methods in Nonlinear Control Theory*. Ed. by Michel Fliess and Michiels Hazewinkel. Dordrecht: Reidel, 1986.

[33] Maison Clouatre, Makhin Thitsa, Michel Fliess, and Cédric Join. « A robust but easily implementable remote control for quadrotors: Experimental acrobatic flight tests ». In: (2020). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2008.00681.

[34] R. Cristi, F.A. Papoulias, and A.J. Healey. « Adaptive sliding mode control of autonomous underwater vehicles in the dive plane ». In: *IEEE Journal of Oceanic Engineering* 15.3 (1990), pp. 152–160.

[35] Ja Z. Cypkin. *Théorie des asservissements par plus-ou-moins*. (Translated from Russian). Paris: Dunod, 1962.

[36] Loïck Degorre, Olivier Chocron, and Emmanuel Delaleau. « Une approche générique de la commande basée modèle des AUV sous-actionnés ». fr. In: *25e Congrès Français de Mécanique, Nantes*. 2022.

[37] E. Delaleau and W. Respondek. « Removing input derivatives and lowering their orders in generalized state-space representations ». In: *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*. Tucson, AZ, USA: IEEE, 1992, pp. 3663–3668. DOI: 10.1109/CDC.1992.370967.

[38] E. Delaleau and J. Rudolph. « Control of flat systems by quasi-static feedback of generalized states ». en. In: *International Journal of Control* 71.5 (Jan. 1998), pp. 745–765. DOI: 10.1080/002071798221551.

[39] Emmanuel Delaleau and Veit Hagenmeyer. « Commande prédictive non linéaire fondée sur la platitude du moteur à induction: application au positionnement de précision ». In: *Journal européen des systèmes automatisés* 36.5 (2002), pp. 737–748.

[40] Taha Elmokadem, Mohamed Zribi, and Kamal Youcef-Toumi. « Trajectory tracking sliding mode control of underactuated AUVs ». en. In: *Nonlinear Dyn* 84.2 (2016), pp. 1079–1091.

[41] S. V. Emelyanov. « A technique to develop complex control by using only the error signal of control variable and its first derivatives ». In: *Automatika i Telemekhanika* 18.10 (1957), pp. 873–885.

[42] S. V. Emelyanov. *Variable Structure Control systems*. (in Russian). Moscow: Nauka, 1967.

[43] Henrique Fagundes Gasparoto, Olivier Chocron, Mohamed Benbouzid, and Pablo Siqueira Meirelles. « Advances in Reconfigurable Vectorial Thrusters for Adaptive Underwater Robots ». en. In: *JMSE* 9.2 (Feb. 2021), p. 170. DOI: 10.3390/jmse9020170.

[44] Henrique Fagundes Gasparoto, Olivier Chocron, Mohamed Benbouzid, Pablo Siqueira Meirelles, and Luiz Saraiva Ferreira. « Torque Analysis of a Flat Reconfigurable Magnetic Coupling Thruster for Marine Renewable Energy Systems Maintenance AUVs ». en. In: *Energies* 12.1 (Dec. 2018), p. 56. DOI: 10.3390/en12010056.

[45] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. (First version published as “Differential equations with discontinuous right par”. Mathematical Proceedings I, 1961, in Russian). Dordrecht: Springer, 1988.

[46] O.-E. Fjellstad and T.I. Fossen. « Position and attitude tracking of AUV's: a quaternion feedback approach ». In: *IEEE Journal of Oceanic Engineering* 19.4 (1994), pp. 512–518.

[47] M. Fliess. « Generalized controller canonical form for linear and nonlinear dynamics ». In: *IEEE Trans. Automat. Contr.* 35.9 (Sept. 1990), pp. 994–1001. DOI: 10.1109/9.58527.

[48] Michel Fliess and Cédric Join. « An alternative to proportional-integral and proportional-integral-derivative regulators: Intelligent proportional-derivative regulators ». en. In: *Intl J Robust & Nonlinear* 32.18 (Dec. 2022), pp. 9512–9524. DOI: 10.1002/rnc.5657.

[49] Michel Fliess and Cédric Join. « Model-free control ». In: *International Journal of Control* 86.12 (Dec. 2013). arXiv:1305.7085 [math], pp. 2228–2252. DOI: 10.1080/00207179.2013.810345.

[50] Michel Fliess and Cédric Join. « MODEL-FREE CONTROL AND INTELLIGENT PID CONTROLLERS: TOWARDS A POSSIBLE TRIVIALIZATION OF NONLINEAR CONTROL? » en. In: *IFAC Proceedings Volumes* 42.10 (2009), pp. 1531–1550. DOI: 10.3182/20090706-3-FR-2004.00256.

[51] Michel Fliess, Jean Lévine, Philippe Martin, François Olivier, and Pierre Rouchon. « Flatness and dynamic feedback linearizability: two approaches ». In: *3rd European Control Conference (ECC)*. Rome, Italy, 1995.

[52] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. « A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems ». In: *IEEE Transactions on Automatic Control* 44 (1999), pp. 922–937.

[53] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. « Flatness and defect of non-linear systems: introductory theory and examples ». en. In: *International Journal of Control* 61.6 (1995), pp. 1327–1361.

[54] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. « Flatness and motion planning: the car with n trailers ». In: *2nd European Control Conference (ECC)*. Groningen, The Netherlands, 1993, pp. 1518–1522.

[55] Michel Fliess, Jean Lévine, Philippe Martin, and Pierre Rouchon. « Sur les systèmes non linéaires différentiellement plats ». In: *Comptes rendus à l'Académie des sciences de Paris* Série I, tome 315 (1992), pp. 619–624.

[56] Michel Fliess, Jean Lévine, and Pierre Rouchon. « Asimplified approach of crane control via generalized state-space model ». In: *30th IEEE Conference on Decision and Control (CDC)*. Brighton, England, 1991, pp. 736–741.

[57] Michel Fliess, Richard Marquez, Emmanuel Delaleau, and Hebertt Sira–Ramírez. « Correcteurs proportionnels-intégraux généralisés ». In: *ESAIM: COCV* 7 (2002), pp. 23–41. DOI: 10.1051/cocv:2002002.

[58] Michel Fliess and Hebertt Sira–Ramírez. « An algebraic framework for linear identification ». In: *ESAIM: COCV* 9 (Jan. 2003), pp. 151–168. DOI: 10.1051/cocv:2003008.

[59] Irmgard Flügge-Lotz. *Discontinuous Automatic Control*. Princeton University Press, 1953.

[60] Irmgard Flügge-Lotz, C. F. Taylor, and H. E. Lindberg. *Investigations on Nonlinear Control*. Patent NACA-TR-1391. 1958.

[61] Thor I. Fossen. « An Adaptive Line-of-Sight (ALOS) Guidance Law for Path Following of Aircraft and Marine Craft ». In: *IEEE Transactions on Control Systems Technology* (2023). Conference Name: IEEE Transactions on Control Systems Technology, pp. 1–8. DOI: 10.1109/TCST.2023.3259819.

[62] Thor I. Fossen. « An Amplitude-Phase Representation of the North-East-Down Kinematic Differential Equations ». In: *IEEE Access* 11 (2023). Conference Name: IEEE Access, pp. 12587–12593. DOI: 10.1109/ACCESS.2023.3242331.

[63] Thor I. Fossen. *Guidance and control of ocean vehicles*. Chichester, NY: Wiley, 1994.

[64] Thor I. Fossen. *Handbook of marine craft hydrodynamics and motion control*. eng. Second edition. Hoboken, NJ Chichester, West Sussex: Wiley, 2021.

[65] Thor I. Fossen. *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. eng. 3rd. Trondheim: Marine Cybernetics, 2002.

[66] Thor I. Fossen, Morten Breivik, and Roger Skjetne. « Line-of-sight path following of underactuated marine craft ». In: *IFAC Proceedings Volumes* 36.21 (2003), pp. 211–216.

[67] Thor I. Fossen and Tor A. Johansen. « A Survey of Control Allocation Methods for Ships and Underwater Vehicles ». In: *14th Mediterranean Conference on Control and Automation*. Ancona, Italy, 2006, pp. 1–6.

[68] Thor I. Fossen and Anastasios M. Lekkas. « Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents ». en. In: *International Journal of Adaptive Control and Signal Processing* 31.4 (2017), pp. 445–463.

[69] Thor I. Fossen and Kristin Y. Pettersen. « On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws ». en. In: *Automatica* 50.11 (2014), pp. 2912–2917.

[70] T.I. Fossen and T. Perez. « Marine Systems Simulator (MSS) - Matlab Toolbox ». en. In: (2004).

[71] T.I. Fossen and S.I. Sagatun. « Adaptive control of nonlinear underwater robotic systems ». In: *IEEE International Conference on Robotics and Automation (ICRA)*. 1991, 1687–1694 vol.2.

[72] Nicolas Gartner, Mathieu Richier, Claire Dune, and Vincent Hugel. « Hydrodynamic Parameters Estimation Using Varying Forces and Numerical Integration Fitting Method ». In: *IEEE Robot. Autom. Lett.* 7.4 (2022), pp. 11713–11719.

[73] Henrique Fagundes Gasparoto, Olivier Chocron, Mohamed Benbouzid, and Pablo Siqueira Meirelles. « Magnetic design and analysis of a radial reconfigurable magnetic coupling thruster for vectorial AUV propulsion ». In: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. Beijing: IEEE, Oct. 2017, pp. 2876–2881. DOI: 10.1109/IECON.2017.8216485.

[74] Scott B. Gibson, Brian McCarter, Daniel J. Stilwell, and Wayne L. Neu. « A comparison of hydrodynamic damping models using least-squares and adaptive identifier methods for autonomous underwater vehicles ». In: *OCEANS 2015 - MTS/IEEE Washington*. Oct. 2015, pp. 1–7. DOI: 10.23919/OCEANS.2015.7401981.

[75] M.N. Grishchenko, V.N. Ivanov, and V. F. Mavritayn. « Multivariable Variable Structure Control Systems ». In: *Proc. 3rd All-Union Conference Automatic Control*. 1958.

[76] J. Guerrero, J. Torres, V. Creuze, and A. Chemori. « Trajectory tracking for autonomous underwater vehicle: An adaptive approach ». en. In: *Ocean Engineering* 172 (Jan. 2019), pp. 511–522. DOI: 10.1016/j.oceaneng.2018.12.027.

[77] J. Guerrero, J. Torres, V. Creuze, A. Chemori, and E. Campos. « Saturation based nonlinear PID control for underwater vehicles: Design, stability analysis and experiments ». en. In: *Mechatronics* 61 (Aug. 2019), pp. 96–105. DOI: 10.1016/j.mechatronics.2019.06.006.

[78] J. Guo, F. C. Chiu, and C. C. Huang. « Design of a sliding mode fuzzy controller for the guidance and control of an autonomous underwater vehicle ». en. In: *Ocean Engineering* 30.16 (Nov. 2003), pp. 2137–2155. DOI: 10.1016/S0029-8018(03)00048-9.

[79] Pierre-Antoine Gédouin, Cédric Join, Emmanuel Delaleau, Jean-Matthieu Bourgeot, Shabnam Arbab Chirani, and Sylvain Calloch. « A new control strategy for shape memory alloys actuators ». In: *ESOMAT 2009 - 8th European Symposium on Martensitic Transformations*. Prague, Czech Republic: EDP Sciences, 2009, p. 07007. DOI: 10.1051/esomat/200907007.

[80] V. Hagenmeyer and E. Delaleau. « Continuous-time non-linear flatness-based predictive control: an exact feedforward linearisation setting with an induction drive example ». In: *Internat. J. Control* 81.10 (2008), pp. 1645–1663.

[81] V. Hagenmeyer and E. Delaleau. « Robustness Analysis of Exact Feedforward Linearization based on differential flatness ». In: *Automatica* 39 (2003), pp. 1941–1946.

[82] V. Hagenmeyer and E. Delaleau. « Robustness Analysis with Respect to Exogenous Perturbations for Flatness-Based Exact Feedforward Linearization ». In: *IEEE Transactions on Automatic and Control* 55.3 (2010), pp. 727–731. URL: https://hal.archives-ouvertes.fr/hal-00431712/document.

[83] Veit Hagenmeyer. *Robust nonlnear tracking controlbased on differential flatness*. Vol. 8. Fortschritt-Berichte. Düsseldorf, Germany: VDI Verlag, 2003.

[84] Veit Hagenmeyer and Emmanuel Delaleau. « Exact feedforward linearization based on differential flatness ». en. In: *International Journal of Control* 76.6 (2003), pp. 537–556.

[85] Veit Hagenmeyer and Emmanuel Delaleau. « Robustness analysis of exact feedforward linearization based on differential flatness ». en. In: *Automatica* 39.11 (2003), pp. 1941–1946.

[86] Veit Hagenmeyer and Marcus Nohr. « Flatness-based two-degree-of-freedom control of industrial semi-batch reactors ». In: *Control and Observer Design for Nonlinear Finite and Infinite Dimensional Systems*. Berlin, Heildelberg: Springer, 2005.

[87] A.J. Healey and D. Lienard. « Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles ». In: *IEEE Journal of Oceanic Engineering* 18.3 (1993), pp. 327–339.

[88] D.E. Humphreys and K.W. Watkinson. *Prediction of Acceleration Hydrodynamic Coefficients for Underwater Vehicles from Geometric Parameters*. Defense Technical Information Center, Feb. 1978.

[89] Frederick H. Imlay. *The Complete Expressions for "Added Mass" of a Rigid Body Moving in an Ideal Fluid*. Research and Development Report 1528. Departement of the Navy, Hydromechanics Laboratory, 1961.

[90] Alberto Isidori. *Nonlinear Control Systems*. Ed. by E. D. Sontag, M. Thoma, A. Isidori, and J. H. Van Schuppen. Communications and Control Engineering. London: Springer London, 1995. DOI: 10.1007/978-1-84628-615-5.

[91] Alberto Isidori. *Nonlinear Control Systems: An Introduction*. Second edition. Berlin: Springer-Verlag, 1989.

[92] Alberto Isidori and A. Ruberti. « On the synthesis of linear input-output responses for nonlinear systems ». In: *Systems & Control Letters* 4 (1984), pp. 17–22.

[93] U. Itkis. *Control Systems of Variable Structures*. New-York: Wiley, 1976.

[94] Bronislav Jakubczyk and Witold Respondek. « On linearization of Control Systems ». In: *Bulletin de l'Académie Polonaises de Sciences* 28.9–10 (1980), pp. 517–522.

[95] H.J. Jayakrishnan. « Position and Attitude control of a Quadrotor UAV using Super Twisting Sliding Mode ». en. In: *IFAC-PapersOnLine* 49.1 (2016), pp. 284–289.

[96] George C. Karras, Charalampos P. Bechlioulis, Sharad Nagappa, Narcis Palomeras, Kostas J. Kyriakopoulos, and Marc Carreras. « Motion control for autonomous underwater vehicles: A robust model ». In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 6529–6534. DOI: 10.1109/ICRA.2014.6907822.

[97] Hassan K. Khalil. *Nonlinear systems*. 3rd ed. Upper Saddle River, N.J: Prentice Hall, 2002.

[98] Efstathios Konstantinidis. « Added mass of a circular cylinder oscillating in a free stream ». In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 469.2156 (Aug. 2013). Publisher: Royal Society, p. 20130135. DOI: 10.1098/rspa.2013.0135.

[99] W. Korobov. « Controllability, stability of some nonlinear systems ». In: *Differencialnyje Uravnienje* 9 (1973), pp. 466–469.

[100] Alexandr I. Korotkin. « Added Masses of Three-Dimensional Bodies in Infinite Fluid ». en. In: *Added Masses of Ship Structures*. Vol. 88. ISSN: 0926-5112 Series Title: Fluid Mechanics and Its Applications. Dordrecht: Springer Netherlands, 2009, pp. 81–102. DOI: 10.1007/978-1-4020-9432-3_3.

[101] Horace Lamb. *Hydrodynamics*. eng. 6th. (Unabridged and unaltered republication of the 1932 ed., Cambridge). New York: Dover, 2005.

[102] Lionel Lapierre and Bruno Jouvencel. « Robust Nonlinear Path-Following Control of an AUV ». In: *IEEE Journal of Oceanic Engineering* 33.2 (2008), pp. 89–102.

[103] Lionel Lapierre and Didik Soetanto. « Nonlinear path-following control of an AUV ». en. In: *Ocean Engineering* 34.11-12 (Aug. 2007), pp. 1734–1744. DOI: 10.1016/j.oceaneng.2006.10.019.

[104] R. K. Lea, R. Allen, and S. L. Merry. « A comparative study of control techniques for an underwater flight vehicle ». en. In: *International Journal of Systems Science* 30.9 (1999), pp. 947–964.

[105] Anastasios Lekkas and Thor Fossen. « Line-of-Sight Guidance for Path Following of Marine Vehicles ». In: June 2013.

[106] Anastasios M. Lekkas and Thor I. Fossen. « Minimization of cross-track and along-track errors for path tracking of marine underactuated vehicles ». In: *2014 European Control Conference (ECC)*. 2014, pp. 3004–3010.

[107] N.E. Leonard. « Control synthesis and adaptation for an underactuated autonomous underwater vehicle ». In: *IEEE Journal of Oceanic Engineering* 20.3 (1995), pp. 211–220.

[108] Arie Levant. « Higher-order sliding modes, differentiation and output-feedback control ». en. In: *International Journal of Control* 76.9-10 (Jan. 2003), pp. 924–941. DOI: 10.1080/0020717031000099029.

[109] Arie Levant. « Sliding order and sliding accuracy in sliding mode control ». en. In: *International Journal of Control* 58.6 (1993), pp. 1247–1263.

[110] Hongliang Li, Hao Chen, Ning Gao, Nadia Aït-Ahmed, Jean-Frederic Charpentier, and Mohamed Benbouzid. « Ship Dynamic Positioning Control Based on Active Disturbance Rejection Control ». In: *Journal of Marine Science and Engineering* 10.7 (2022). DOI: 10.3390/jmse10070865.

[111] Wenkui Li, Zhu Zhou, Jianlu Lou, and Xuguang Zhang. « A 3D trajectory tracking algorithm for AUV ». In: *J. Phys.: Conf. Ser.* 1873.1 (2021), p. 012055.

[112] Song Liang, Fei Wu, Guangyi Zhang, Xiaoqiang Dai, and Qingjun Zeng. « Multivariable fast nonsingular terminal sliding mode control for remotely operated vehicle ». In: *2017 36th Chinese Control Conference (CCC)*. Dalian, China: IEEE, July 2017, pp. 3817–3823. DOI: 10.23919/ChiCC.2017.8027954.

[113] P. S. Londhe, Dinesh. D. Dhadekar, B. M. Patre, and L. M. Waghmare. « Non-singular terminal sliding mode control for robust trajectory tracking control of an autonomous underwater vehicle ». In: *Indian Control Conference (ICC)*. Guwahati, India, 2017, pp. 443–449.

[114] P. S. Londhe and B. M. Patre. « Adaptive fuzzy sliding mode control for robust trajectory tracking control of an autonomous underwater vehicle ». en. In: *Intel Serv Robotics* 12.1 (Jan. 2019), pp. 87–102. DOI: 10.1007/s11370-018-0263-z.

[115] Antonio Loria. « From feedback to cascade-interconnected systems: Breaking the loop ». In: *2008 47th IEEE Conference on Decision and Control*. Cancun, Mexico: IEEE, 2008, pp. 4109–4114. DOI: 10.1109/CDC.2008.4738647.

[116] Jean Lévine. *Analysis and Control of Nonlinear Systems: A flatness-based approach*. Berlin, Heildelberg, Germany: Springer-Verlag, 2009.

[117] Man Zhihong and Xing Huo Yu. « Terminal sliding mode control of MIMO linear systems ». In: *IEEE Transactions on Circuits Systems I* 44.11 (1997), pp. 1065–1070.

[118] Stephen C. Martin and Louis L. Whitcomb. « Nonlinear Model-Based Tracking Control of Underwater Vehicles With Three Degree-of-Freedom Fully Coupled Dynamical Plant Models: Theory and Experimental Evaluation ». In: *IEEE Transactions on Control Systems Technology* 26.2 (2018), pp. 404–414.

[119] Andreas B. Martinsen, Anastasios M. Lekkas, and Sebastien Gros. « Autonomous docking using direct optimal control ». en. In: *IFAC-PapersOnLine* 52.21 (2019), pp. 97–102. DOI: 10.1016/j.ifacol.2019.12.290.

[120] Mamadou Mboup, Cédric Join, and Michel Fliess. « Numerical differentiation with annihilators in noisy environment ». en. In: *Numer Algor* 50.4 (Apr. 2009), pp. 439–467. DOI: 10.1007/s11075-008-9236-1.

[121] Timothy W. McLain and Stephen M. Rock. « Development and Experimental Validation of an Underwater Manipulator Hydrodynamic Model ». en. In: *The International Journal of Robotics Research* 17.7 (July 1998), pp. 748–759. DOI: 10.1177/027836499801700705.

[122] Alasdair Mitchell, Euan McGookin, and David Murray-Smith. « Comparison of Control Methods for Autonomous Underwater Vehicles ». en. In: *IFAC Proceedings Volumes*. IFAC Workshop on Guidance and Control of Underwater Vehicles 2003, Newport, South Wales, UK, 9-11 April 2003 36.4 (2003), pp. 37–42.

[123] Marcos Moreno-Gonzalez, Antonio Artuñedo, Jorge Villagra, Cédric Join, and Michel Fliess. « Speed-Adaptive Model-Free Path-Tracking Control for Autonomous Vehicles: Analysis and Design ». en. In: *Vehicles* 5.2 (June 2023), pp. 698–717. DOI: 10.3390/vehicles5020038.

[124] P. Muir and C. Neuman. « Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot ». In: *Proceedings. 1987 IEEE International Conference on Robotics and Automation*. Vol. 4. 1987, pp. 1772–1778. DOI: 10.1109/ROBOT.1987.1087767.

[125] Duy Nguyen-Tuong, Matthias Seeger, and Jan Peters. « Computed torque control with nonparametric regression models ». In: *IEEE American Control Conference (ACC)*. Seattle, WA, 2008, pp. 212–217.

[126] Henk Nijmeijer and Arjan J. van der Schaft. *Nonlinear Dynamical Control Systems*. New York: Springer-Verlag, 1990.

[127] Brigitte d'Andrea Novel, Michel Fliess, Cedric Join, Hugues Mounier, and Bruno Steux. « A mathematical explanation via “intelligent” PID controllers of the strange ubiquity of PIDs ». In: *18th Mediterranean Conference on Control and Automation, MED'10*. Marrakech, Morocco: IEEE, June 2010, pp. 395–400. DOI: 10.1109/MED.2010.5547700.

[128] Romeo Ortega and Mark W. Spong. « Adaptive motion control of rigid robots: A tutorial ». en. In: *Automatica* 25.6 (1989), pp. 877–888.

[129] R. Penrose. « A generalized inverse for matrices ». en. In: *Math. Proc. Camb. Phil. Soc.* 51.3 (July 1955), pp. 406–413. DOI: 10.1017/S0305004100030401.

[130] R. Pepy, A. Lambert, and H. Mounier. « Path Planning using a Dynamic Vehicle Model ». In: *2006 2nd International Conference on Information & Communication Technologies*. Vol. 1. Damascus, Syria: IEEE, 2006, pp. 781–786. DOI: 10.1109/ICTTA.2006.1684472.

[131] K.Y. Pettersen and O. Egeland. « Position and attitude control of an underactuated autonomous underwater vehicle ». In: *Proceedings of 35th IEEE Conference on Decision and Control*. Vol. 1. ISSN: 0191-2216. Dec. 1996, 987–991 vol.1. DOI: 10.1109/CDC.1996.574614.

[132] K.Y. Pettersen and E. Lefeber. « Way-point tracking control of ships ». In: *40th IEEE Conference on Decision and Control (CDC)*. Vol. 1. Orlando, FL, USA, 2001, pp. 940–945.

[133] Philip Polack, Sébastien Delprat, and Brigitte d'Andréa Novel. « Brake and velocity model-free control on an actual vehicle ». en. In: *Control Engineering Practice* 92 (Nov. 2019), p. 104072. DOI: 10.1016/j.conengprac.2019.06.011.

[134] Lei Qiao, Bowen Yi, Defeng Wu, and Weidong Zhang. « Design of three exponentially convergent robust controllers for the trajectory tracking of autonomous underwater vehicles ». en. In: *Ocean Engineering* 134 (Apr. 2017), pp. 157–172. DOI: 10.1016/j.oceaneng.2017.02.006.

[135] Gerasimos G. Rigatos. « Differential Flatness Theory and Flatness-Based Control ». en. In: *Nonlinear Control and Filtering Using Differential Flatness Approaches: Applications to Electromechanical Systems*. Ed. by Gerasimos G. Rigatos. Cham, Switzerland: Springer International Publishing, 2015, pp. 47–101.

[136] Gerasimos G. Rigatos. *Nonlinear filtering and control using differential flatness approaches: Applications to electromechanical systems*. Cham, Switzerland: Springer International Publishing, 2015.

[137] Gerosimo G. Rigatos, G. V. Raffo, and Pierluigi Siano. « AUV Control and Navigation with Differential Flatness Theory and Derivative-Free Nonlinear Kalman Filtering ». en. In: *Intelligent Industrial Systems* 3.1 (2017), pp. 29–41.

[138] Ralf Rothfuss, Joachim Rudolph, and Michael Zeitz. « Flatness based control of a nonlinear chemical reactor model ». In: *Automatica* 32.10 (1996), pp. 1433–1439.

[139] P. Rouchon, M. Fliess, J. Levine, and P. Martin. « Flatness, motion planning and trailer systems ». In: *Proceedings of 32nd IEEE Conference on Decision and Control*. San Antonio, TX, USA: IEEE, 1993, pp. 2700–2705. DOI: 10.1109/CDC.1993.325686.

[140] Pierre Rouchon. « Vibrational control and flatness of chemical reactors ». In: *Proceedings of the Symposium on Control, Optimization and Supervision; Computational Engineering in Systems Applications IMACS Multiconference*. Ed. by P. Borne, M. Staroswiecki, J. P. Cassar, and S. El Khattabi. Lille, France, 1996, pp. 211–212.

[141] Spandan Roy, Sambhunath Nandy, Sankar Nath Shome, and Ranjit Ray. « Robust position control of an autonomous underwater vehicle: A comparative study ». In: *2013 IEEE International Conference on Automation Science and Engineering (CASE)*. ISSN: 2161-8089. Aug. 2013, pp. 1002–1007. DOI: 10.1109/CoASE.2013.6653951.

[142] Joachim Rudolph. *Flatness-Based Control: an Introduction*. Düren, Germany: Shaker Verlag, 2021.

[143] Taavi Salumae, Ahmed Chemori, and Maarja Kruusmaa. « Motion Control of a Hovering Biomimetic Four-Fin Underwater Robot ». In: *IEEE J. Oceanic Eng.* 44.1 (Jan. 2019), pp. 54–71. DOI: 10.1109/JOE.2017.2774318.

[144] Albert Sans-Muntadas, Eleni Kelasidi, Kristin Y. Pettersen, and Edmund Brekke. « Learning an AUV docking maneuver with a convolutional neural network ». en. In: *IFAC Journal of Systems and Control* 8 (June 2019), p. 100049. DOI: 10.1016/j.ifacsc.2019.100049.

[145] Arjan J. van der Schaft. « Linearization and input-ouput decoupling for general nonlinear systems ». In: *Systems & Control Letters* 5 (1984), pp. 27–33.

[146] Guido Schillaci, Fabio Schillaci, and Verena V. Hafner. « A Customisable Underwater Robot ». In: *arXiv:1707.06564 [cs]* (July 2017). arXiv: 1707.06564.

[147] Eduardo Sebastián and Miguel A. Sotelo. « Adaptive Fuzzy Sliding Mode Controller for the Kinematic Variables of an Underwater Vehicle ». en. In: *J Intell Robot Syst* 49.2 (May 2007), pp. 189–215. DOI: 10.1007/s10846-007-9144-y.

[148] K. Shigin Ye. « On improvement of Transient Processes in Variable Parameter Correcting Elements ». In: *Avtomatika i Telemekhanika* 19.4 (1958), pp. 306–312.

[149] Anmol Shrivastava, M. Karthikeyan, and Prabhu Rajagopal. « Modelling and Motion Control of an Underactuated Autonomous Underwater Vehicle ». In: *6th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. Tokyo, Japan, 2021, pp. 62–68.

[150] Yuri Shtessel, Christopher Edwards, Leonid Fridman, and Arie Levant. *Sliding Mode Control and Observation*. Control Engineering. New York, NY: Springer New York, 2014. DOI: 10.1007/978-0-8176-4893-0.

[151] Yuri B. Shtessel, Jaime A. Moreno, Franck Plestan, Leonid M. Fridman, and Alexander S. Poznyak. « Super-twisting adaptive sliding mode control: A Lyapunov design ». In: *49th IEEE Conference on Decision and Control (CDC)*. Atlanta, GA, USA: IEEE, Dec. 2010, pp. 5109–5113. DOI: 10.1109/CDC.2010.5717908.

[152] Paulo Sergio Pereira da Silva and Emmanuel Delaleau. « Algebraic necessary and sufficient conditions of input-output linearization ». In: *Forum Mathematicum* 13.3 (2001), pp. 335–357.

[153] Hebertt Sira-Ramírez. *Sliding Mode Control: The Delta-Sigma Modulation Approach*. Cham: Birkhäuser, 2015.

[154] Hebertt Sira-Ramírez and Sunil Kumar Agrawal. *Differentially flat systems*. eng. New York, Basel: Marcel Dekker, 2004.

[155] Jean-Jacques E. Slotine. « Tracking Control of Nonlinear Systems using Sliding Surfaces ». PhD thesis. Cambridge, MA: Massachusetts Institute of Technology, 1983.

[156] Jean-Jacques E. Slotine and Weiping Li. *Applied nonlinear control*. eng. Upper Saddle River, NJ: Prentice Hall, 1991.

[157] Jean-Jacques E. Slotine and Shankar S. Sastry. « Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators ». In: *International Journal of Control* 38.2 (1983), pp. 465–492.

[158] D.A. Smallwood and L.L. Whitcomb. « Model-based dynamic positioning of underwater robotic vehicles: theory and experiment ». In: *IEEE Journal of Oceanic Engineering* 29.1 (2004), pp. 169–186.

[159] Feijun Song and S.M. Smith. « Design of sliding mode fuzzy controllers for an autonomous underwater vehicle without system model ». In: *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*. Vol. 2. Sept. 2000, 835–840 vol.2. DOI: 10.1109/OCEANS.2000.881362.

[160] Mark W. Spong. « Underactuated Mechanical Systems ». In: (June 2005). DOI: 10.1007/BFb0015081.

[161] M.W. Spong. « Partial feedback linearization of underactuated mechanical systems ». In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*. Vol. 1. Munich, Germany: IEEE, 1994, pp. 314–321. DOI: 10.1109/IROS.1994.407375.

[162] R. Su. « On the linear equivalents of nonlinear systems ». In: *Systems & Control Letters* 2 (1982), pp. 48–52.

[163] Russ Tedrake. *Underactuated Robotics. Algorithms for Walking, Running, Swimming, Flying, and Manipulation*. 2023. URL: https://underactuated.csail.mit.edu.

[164] Auwal Shehu Tijjani, Ahmed Chemori, and Vincent Creuze. « A survey on tracking control of unmanned underwater vehicles: Experiments-based approach ». en. In: *Annual Reviews in Control* 54 (2022), pp. 125–147. DOI: 10.1016/j.arcontrol.2022.07.001.

[165] V. I. Utkin. « Discontinuous Control Systems: State of Art in Theory and Application ». In: *IFAC Proceedings Volumes* 20.5 (1987), pp. 25–44.

[166] V. I. Utkin. *Sliding Modes and their Applications in Variable Structure System*. Moscow: Ed. MIR, 1978.

[167] Pierre Varignon. *Eléments de Mathématiques*. Brunet. Paris, 1731.

[168] Emanuel P. Vega, Olivier Chocron, Janito Vequiero Ferreira, Mohamed Benbouzid, and Pabo Siquiera Meirelles. « Evaluation of AUV Fixed and Vectorial Propulsion Systems with Dynamic Simulation and Non-linear Control ». In: *41st Annual Conference of the IEEE Industrial Electronics Society (IECON2015)*. Yokohama, Japan, 2015.

[169] Emanuel Pablo Vega, Olivier Chocron, and Mohamed El Hachemi Benbouzid. « AUV Propulsion Systems Modeling Analysis ». In: *IREMOS* 7.5 (Oct. 2014), p. 827. DOI: 10.15866/iremos.v7i5.3648.

[170] J. Villagra, B. d'Andrea Novel, H. Mounier, and M. Pengov. « Flatness-Based Vehicle Steering Control Strategy With SDRE Feedback Gains Tuned Via a Sensitivity Approach ». In: *IEEE Trans. Contr. Syst. Technol.* 15.3 (May 2007), pp. 554–565. DOI: 10.1109/TCST.2007.894651.

[171] A. Wahl and E. D. Gilles. « Model predictive versus linear quadratic control for the tracking problem of automatic river navigation ». In: *5th European Control Conference (ECC)*. Karlsruhe, 1999, pp. 1137–1142.

[172] Guoqing Xia, Yu Zhang, Wei Zhang, Kaihang Zhang, and Haoyu Yang. « Robust adaptive super-twisting sliding mode formation controller for homing of multi-underactuated AUV recovery system with uncertainties ». en. In: *ISA Transactions* (2022), S0019057822001719.

[173] Xianbo Xiang, Lionel Lapierre, and Bruno Jouvencel. « Smooth transition of AUV motion control: From fully-actuated to under-actuated configuration ». en. In: *Robotics and Autonomous Systems* 67 (2015), pp. 14–22.

[174] Toya Yamada, Hiroshi Kinjo, Kunihiko Nakazono, Naoki Oshiro, and Eiho Uezato. « Fuzzy controller for AUV robots based on machine learning and genetic algorithm ». en. In: *Artif Life Robotics* 28.3 (Aug. 2023), pp. 632–641. DOI: 10.1007/s10015-023-00881-z.

[175] D. Yoerger and J.-J. E. Slotine. « Robust trajectory control of underwater vehicles ». In: *IEEE Journal of Oceanic Engineering* 10.4 (1985), pp. 462–470.

[176] D. Yoerger and Jean-Jacques E. Slotine. « Nonlinear Trajectory Control of Autonomous Underwater Vehicule Using the Sliding Methodology ». In: *IEEE Conference OCEANS*. 1984.

[177] D.R. Yoerger and J.-J.E. Slotine. « Adaptive sliding control of an experimental underwater vehicle ». In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. Sacramento, CA, USA: IEEE Comput. Soc. Press, 1991, pp. 2746–2751. DOI: 10.1109/ROBOT.1991.132047.

[178] K.D. Young, V.I. Utkin, and U. Ozguner. « A control engineer's guide to sliding mode control ». In: *IEEE Transactions on Control Systems Technology* 7.3 (1999), pp. 328–342.

[179] Junzhi Yu, Jincun Liu, Zhengxing Wu, and Hao Fang. « Depth Control of a Bioinspired Robotic Dolphin Based on Sliding-Mode Fuzzy Control Method ». In: *IEEE Transactions on Industrial Electronics* 65.3 (2018), pp. 2429–2438.

[180] Junzhi Yu, Zongshuai Su, Zhengxing Wu, and Min Tan. « Development of a Fast-Swimming Dolphin Robot Capable of Leaping ». In: *IEEE/ASME Trans. Mechatron.* 21.5 (Oct. 2016), pp. 2307–2316. DOI: 10.1109/TMECH.2016.2572720.

[181] R. Yu, Q. Zhu, G. Xia, and Z. Liu. « Sliding mode tracking control of an underactuated surface vessel ». en. In: *IET Control Theory Appl.* 6.3 (2012), p. 461.

[182] J. Yuh, Jing Nie, and C. S. G. Lee. « Experimental study on adaptive control of underwater robots ». In: *IEEE International Conference on Robotics and Automation (ICRA)*. 1999, pp. 393–398.

[183] Side Zhao and J. Yuh. « Experimental study on advanced underwater robot control ». In: *IEEE Transactions on Robotics* 21.4 (Aug. 2005). Conference Name: IEEE Transactions on Robotics, pp. 695–703. DOI: 10.1109/TRO.2005.844682.

[184] V. V. Zharinov. *Geometrical aspect of partial differential equations*. Singapore: World Scientific, 1992.

[185] Jiajia Zhou, Xinyi Zhao, Tao Chen, Zheping Yan, and Zewen Yang. « Trajectory Tracking Control of an Underactuated AUV Based on Backstepping Sliding Mode With State Prediction ». In: *IEEE Access* 7 (2019). Conference Name: IEEE Access, pp. 181983–181993. DOI: 10.1109/ACCESS.2019.2958360.

COLLEGE DOCTORAL BRETAGNE SCIENCES POUR L'INGENIEUR

Institut de Recherche Dupuy de Lôme
UMR CNRS 6027

ÉCOLE NATIONALE D'INGÉNIEURS DE BREST

Titre : Analyse et commande des robots sous-marins autonomes à propulsion vectorielle reconfigurable.

Mot clés : Guidage, Contrôle, AUV sous-actionnés, Propulsion Vectorielle

Résumé : Ce manuscrit présente des avancées en matière de contrôle et de conception de véhicules marins autonomes. Il se concentre sur l'amélioration de la mobilité des véhicules marins afin de débloquer de nouvelles applications telles que l'inspection et la maintenance des systèmes de production d'énergie verte en mer. Les deux axes principaux de ce travail sont le contrôle des véhicules marins sous-actionnés et la conception de nouvelles solutions de propulsion vectorielle. Deux nouveaux contrôleurs ont été mis au point pour les véhicules sous-actionnés. Le premier contrôleur repose sur un nouveau principe de guidage cinématique. Une nouvelle matrice de gain non diagonale basée sur le modèle, appelée $\mathcal{H}$, est introduite pour exploiter les relations cinématiques du modèle. Elle permet de contrôler un véhicule dans une tâche de suivi de chemin. Le deuxième contrôleur présenté dans ce travail est basé sur la platitude différentielle. Il est calculé sur une version plate du modèle de navire de surfac. Le contrôleur est associé à un iPD pour une plus grande robustesse et un meilleur comportement. La loi de commande est adaptée à la commande des véhicules de surface. Un nouveau système de poussée vectorielle basé sur deux propulseurs parallèles montés sur un anneau rotatif est également présenté pour améliorer la mobilité des AUV sans créer de nouveaux effets de couplage.

Title: Analysis and control of autonomous underwater vehicles with reconfigurable vectoring thrust.

Keywords: Guidance, Control, Underactuated AUVs, Vectoring Thrust

Abstract: This work presents advances in the control and design of autonomous marine vehicles. It is focused on enhancing the mobility of marine vehicles to unlock new applications like the inspection and maintenance of off-shore green energy production plants or the observation and protection of marine environments. The two main axes of this work are the control of underactuated marine vehicles and the design of new vectoring thrust solutions. Notably, two new controllers have been developed for underactuated marine craft. The first controller relies on a new kinematic guidance principle. A new non diagonal model-based gain matrix denoted $\mathcal{H}$ is introduced to exploit kinematic relations of the model. It is shown to allow control of a torpedo shaped vehicle on a path following task. The second controller presented in this work is based on differential flatness. It is calculated on a flat version of the surface vessel model where the usual defects preventing flatness are canceled. The controller is associated with an iPD for increased robustness and better behavior away from the trajectory. The resulting control law is shown to be suited for the control of a generic surface vessel. A new vectoring thrust system based on two parallel thrusters mounted on a rotating ring is also presented to enhance the mobility of AUVs without creating new coupled effects.