Variable gain contouring control for multi-axis machine tools

Tan Quang Duong

NNT: 2018SACLC009

Thèse de doctorat de l'Université Paris-Saclay préparée à CentraleSupélec

École Doctorale n° 580

Sciences et Technologies de l'Information et de la Communication

Spécialité de doctorat: Automatique

Par

Tan Quang DUONG

Variable gain contouring control for multi-axis machine tools

Thèse présentée et soutenue à Gif-sur-Yvette, le 12 Mars 2018.

Composition du Jury :

M. Emmanuel DUC	Professeur des Universités, SIGMA Clermont	Président
M. Mohamed ABBAS-TURKI	Maître de Conférences, ENS Paris-Saclay	Examinateur
M. Kaan ERKORKMAZ	Professor, University of Waterloo	Rapporteur
M. Richard BÉARÉE	Professeur adjoint, Arts & Métiers ParisTech	Rapporteur
M. Pedro RODRIGUEZ-AYERBE	Professeur des Universités, CentraleSupélec	Directeur de thèse
M. Sylvain LAVERNHE	Professeur adjoint, ENS Paris-Saclay	Co-directeur de thèse
M. Christophe TOURNIER	Professeur des Universités, ENS Paris-Saclay	Co-encadrant
M. Didier DUMUR	Professeur des Universités, CentraleSupélec	Co-encadrant

À Phuong

À ma famille

Acknowledgments

First, I would like to express my special thanks to my supervisors: Professor Pedro RODRIGUEZ-AYERBE and Professor Didier DUMUR at the L2S laboratory, CentraleSupélec; Associate Professor Sylvain LAVERNHE and Professor Christophe TOURNIER at the LURPA laboratory, ENS Paris Saclay. Over my three-year PhD thesis, experiencing a lot of meetings and discussions, you all have given me precious knowledges and excellent supervisions, guiding me to accomplish the thesis. Thank you for your patience, thoroughness, belief, and encouragement. I appreciate deeply all of your supports, not only in the scientific point of view but also for my personal life. Thank you so much!

It is a great honor for me that Professor Kaan ERKORKMAZ at University of Waterloo, Canada and Associate Professor Richard BÉARÉE at Arts & Métiers ParisTech have accepted to be two reporters of my PhD thesis. Thank you for giving me many good comments, excellent proposes and recommendations to improve my thesis and to extend perspectives for future works. I would like also to thank Professor Emmanuel DUC at SIGMA Clermont, and Associate Professor Mohamed ABBAS-TURKI at ENS Paris-Saclay, for accepting to be the jury members of my thesis defense. Thank you so much for your very good questions and comments about my PhD thesis.

It is a great chance for me to thank the L2S and LURPA laboratories, where I have worked in a rich scientific environment, and have enjoyed friendly vivid activities. I would like also to acknowledge the Digiteo foundation for accepting and funding my 3-year PhD thesis.

I would like to say thank you for all of my friends and colleagues: Quang Binh, Lam Thanh, Minh Nguyen, Minh Tri, Son Nano, Dai Hai, Quang Bang, Duy Minh, Abid, Mohammed, Michele, Lorène, Kévin, Kamel, Laureen,... Thanks for your helps, for our good memories, picnics, trips together.

It is not only today that I would like to express my profound gratitude to my parents, who's giving me the best of their lives. Thank you so much! Mommy, Daddy, I love you!

Finally, I dedicate this success for you, Phuong. You are my friend, my wife, my sources of motivation and love. I could not succeed this PhD thesis without you. At this moment, I remember not only our happy moments, our first favorite food "vit om sau", our phone call at

the airport before my first flight to France, our wedding, our trips,... but also the difficult moments that we have overcome together. I appreciate all of them as a part of our adventure. Now, let's write the next beautiful chapters of our adventures. Let's give this gift for our lovely son and for our future children ;). Thank you so much! I love you, my cat!

Contents

List of Figures

List of Tables

List of acronyms and symbols

MT	Machine Tool
HSM	High Speed Machining
CNC	Computer Numerical Control
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
OGA	Off-line Gain Adjustment
GPC	Generalized Predictive Control
FFW	Feed Forward
P	Proportional
PID	Proportional-Integral
PID	Proportional-Integral-Derivative
IKT	Inverse Kinematic Transformation
FKT	Forward Kinematic Transformation
TE	Tracking error
CE	Contour error
$\mathbf{C_C}$	Cutter contact (the contact point between the tool and the finished surface)
$\mathbf{C_E}$	Center point of the tool
$\mathbf{C_L}$	Cutter location described in the NC program, which is generally the extremity point of the tool
$\mathbf{C_H}$	High tool location point in flank milling
H	Distance between $\mathbf{C_L}$ and $\mathbf{C_H}$ in flank milling
K^P	Proportional gain
K^F	Feed forward gain for velocity loop
K^{global}	Global gain
u	Unit vector of the tool axis orientation
r	Radius of the hemispheric tool and small radius of the toric tool
R	Radius of the cylindrical tool and large radius of the toric tool
O	Origin of the programming frame
n	Unit surface normal vector at the tool contact point
f	Unit feed vector at the tool contact point

$\mathbf{t}$	Unit transverse vector at the tool contact point
θ_t	Tilt angle
θ_n	Yaw angle
ε	Contour error variable
ε_L	Contour error variable at the tool location point $\mathbf{C_L}$
ε_C	Contour error variable at the tool contact point $\mathbf{C_C}$
ε_u	Contour error variable for tool orientation axis $\mathbf{u}$
ε_H	Contour error variable at the higher tool location point $\mathbf{C_H}$
ε_F	Contour error variable in flank milling

Résumé

Introduction

Les techniques d'usinage avancées sont indispensables pour le développement des industries manufacturières. L'une de ces techniques, l'usinage à grande vitesse (UGV), est le sujet principal de cette thèse de doctorat. Elle remplace l'usinage à vitesse conventionnelle (UVC) pour la réalisation des formes de pièces complexes dans l'industrie aérospatiale, automobile ou des moules, etc. Le principal avantage de l'usinage à grande vitesse accroit le taux d'enlèvement de matière et améliore la qualité de surface. Par conséquent, il est évident qu'un avantage économique peut être obtenu.

Généralement, de nombreuses étapes interviennent dans l'usinage à grande vitesse, comme le montre la Fig. 1, depuis la conception assistée par ordinateur (CAO), la génération de trajectoire par la fabrication assistée par ordinateur (FAO), les traitements d'exécution par la commande numérique (CN), y compris l'interpolation temporelle et la commande des axes, jusqu'au comportement structurel à l'intérieur de la machine-outil. À la fin de ce processus, on obtient le déplacement réel de l'outil, ainsi que le profil de la vitesse d'avance et le profil d'erreur de contour (CE) le long de la trajectoire d'usinage.

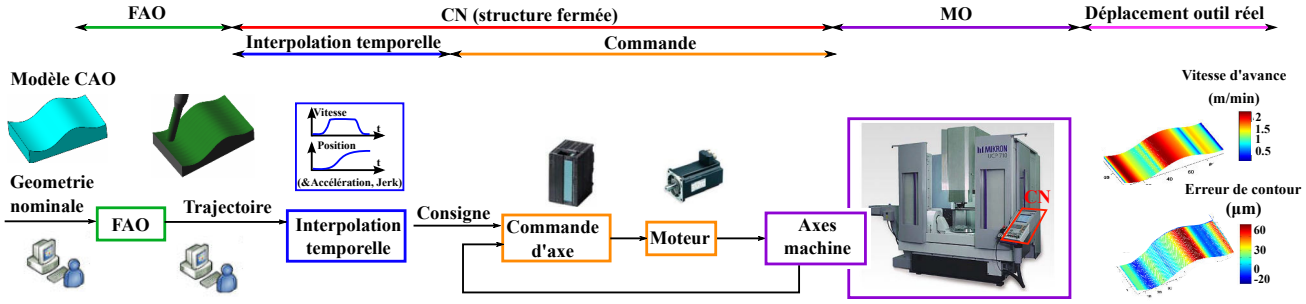

Figure 1: Processus d'usinage à grande vitesse simplifié

En raison de la forte concurrence dans les industries manufacturières, il y a de plus en plus de demandes pour la mise en œuvre des stratégies d'usinage à grande vitesse. Pour augmenter les performances de l'usinage à grande vitesse, ce travail de doctorat se concentre principalement

sur l'amélioration de l'étape de commande d'axe, en particulier la commande en position des axes par la commande numérique. La motivation générale est que si l'action de commande fonctionne bien, la pièce usinée sera très proche de celle désirée. Par conséquent, les étapes de polissage manuelles après usinage peuvent être réduites. La qualité de coupe, la productivité et les avantages économiques sont donc évidents.

Le cadre de ces travaux de recherche est mis en évidence dans la Fig. 2. Le processus est constitué de multiples entrées et sorties.

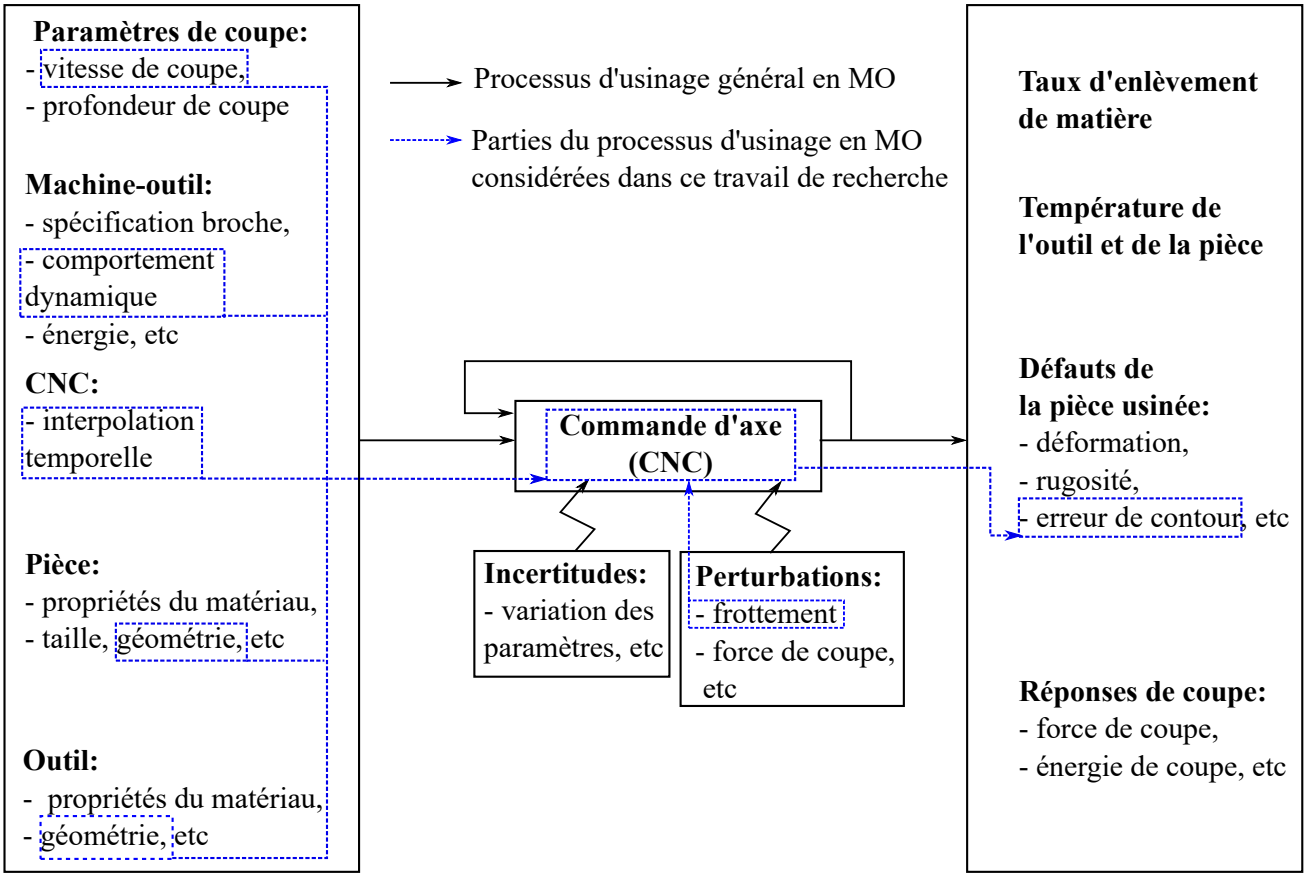

Figure 2: Usinage général en machine outil [Dagiloke 1995] et paramètres d'usinage considérés dans ce travail de recherche

Dans ce processus, les paramètres de coupe (vitesse de coupe, profondeur de coupe, etc), configuration et spécification de la machine-outil (spécification broche, comportement dynamique, etc), de la CN (interpolation temporelle), de la pièce (propriétés du matériau, taille, géométrie, etc) et de l'outil (propriétés du matériau, géométrie, etc) sont des données d'entrée. Tandis que les

effets de sortie sont le taux d'enlèvement de matière, la température de l'outil et de la pièce, les défauts de la pièce usinée, les forces de coupe, etc. La partie centrale du processus d'usinage sur machine outil est la commande d'axe réalisée en boucle fermée, étant sujette à des incertitudes (par exemple la variation des paramètres d'entraînement, etc) et à des perturbations (par exemple frottements, force de coupe, etc). Dans ce travail de recherche, cinq facteurs d'entrée sont principalement pris en compte, notamment la vitesse de coupe, le comportement dynamique de la structure de la machine outil, la vitesse d'avance, la géométrie de la pièce/surface désirée et l'outil. Seul le frottement est considéré comme perturbation de la commande d'axe. En sortie, seul le profil d'erreur de contour est pris en compte.

Pour améliorer les performances de la CN industrielle, les techniques d'interpolation temporelle, le processeur, la motorisation et l'actionneur ont été considérablement améliorés. Toutefois, sa structure et sa stratégie de commande n'ont eu que peu d'améliorations. En fait, chaque entraînement d'axe est contrôlé à travers une structure classique en cascade, comprenant des boucles de position, de vitesse et de courant, respectivement des boucles externes aux boucles internes. Tandis que les deux boucles internes sont contrôlées par des régulateurs proportionnel - intégral (PI), la boucle de position externe est contrôlée par un régulateur Proportionnel (P) combiné avec une action d'anticipation en vitesse (Feed Forward -FFW) pour la boucle de vitesse [Altintas 2000a].

Au sein du processus de commande, il est évident que les petites discontinuités dans la trajectoire peuvent exciter les modes naturels de la structure mécanique, provoquant des vibrations de la machine. Par conséquent, la trajectoire de référence doit d'abord être lissée. Pour ce faire, la planification de la vitesse est réalisée en tenant compte des limites cinématiques de la machine, en particulier les limites de jerk [Erkorkmaz 2001a, Barre 2005, Beudaert 2012]. Par la suite, on s'attend à ce que le contrôleur produise la trajectoire résultante en respectant toutes les contraintes cinématiques de la trajectoire de référence.

Étant donné que l'usinage des surfaces de forme complexe est principalement concerné dans cette étude, il y a de nombreuses raisons qui expliquent qu'avec la commande classique mentionnée ci-dessus, il soit difficile d'obtenir une grande précision de contour. D'une part, dans l'usinage à grande vitesse, la vitesse de coupe élevée conduit à une vitesse d'avance élevée. L'interpolation d'une telle vitesse d'avance augmente également la vitesse, l'accélération et le jerk de chaque axe lors de l'exécution de la trajectoire. Par conséquent, en suivant le profil libre, les axes de la machine-outil peuvent avoir des caractéristiques cinématiques élevées et

des fréquences élevées. D'autre part, la structure de commande d'axe classique a une bande passante limitée. En outre, les sollicitations dynamiques de la machine-outil changent; elle est soumise à des perturbations, telles que des variations d'entraînement dues à la variation de la masse de la pièce et des conditions de lubrification, au frottement et aux forces de coupe [Altintas 2000a, Altintas 2000b, Erkorkmaz 2001b, Tang 2013, Khoshdarregi 2014]. Ces effets augmentent l'erreur de suivi et génèrent une erreur de contour, définie de manière classique comme la distance orthogonale entre le point de contact réel de l'outil et le profil souhaité.

Pour relever ces défis, les chercheurs ont proposé de nombreuses stratégies de commande avancées et intelligentes comme alternatives à la technique classique, telles que Cross Couple Control [Koren 1980, Koren 1991, Zhao 2013], Sliding Mode Control [Altintas 2000b], Adaptive Robust Control [Yao 1997, Davis 2015]. Ces approches sont plus ou moins validées par des résultats de simulation et/ou des essais expérimentaux utilisant une CN ouverte [Altintas 1994, Erkorkmaz 2001b, Beudaert 2014]. Cependant, l'implémentation des contrôleurs avancés dans la version commerciale des CN n'est pas encore réalisée. Ceci est dû au fait que de façon générale, une CN ne permet jusqu'à présent qu'un accès limité au cœur des contrôleurs; elles restent donc "fermées". Une autre raison est que presque toutes les stratégies de commande proposées nécessitent une modification de la structure de commande classique.

En résumé, cette thèse de doctorat "Commande à gains variables de l'erreur de contour pour l'usinage multiaxes" vise à améliorer la précision de contour en usinage multi-axes grande vitesse pour des surfaces de forme complexe, en exploitant toutes les possibilités de la structure de commande classique de l'entraînement d'axes, et en proposant des solutions que les constructeurs pourraient intégrer dans leur CN commerciale. L'amélioration du suivi de la trajectoire signifie que l'erreur de contour doit être contrôlée aussi faible que possible. Dans cette étude, à l'exception du frottement, les autres sources de perturbations et d'incertitudes sont négligées. De plus, compte tenu des vibrations de la machine, l'étape d'interpolation temporelle produit les mouvements d'axes, en respectant les contraintes cinématiques des axes, notamment les limites de jerk. Ces travaux proposent une méthode permettant de générer des gains de commande variables optimaux pour la réduction de l'erreur de contour dans le processus d'usinage. L'optimisation proposée est ici soumise à diverses contraintes que sont les limitations cinématiques des axes (vitesse, accélération et jerk), la stabilité de l'entraînement des axes et les limites de courant du moteur.

Cette thèse de doctorat est réalisée à l'Université de Paris-Saclay, dans le cadre de la coopération

entre deux laboratoires de recherche français: le Laboratoire des Signaux et Systèmes (L2S) de CentraleSupélec et le Laboratoire Universitaire de Recherche en Production Automatisée (LURPA) de l'ENS Paris-Saclay. Ce travail est enregistré sous N° 2014-812D - Projet OMEGA et soutenu par la fondation DIGITEO. La méthode proposée est appliquée au centre d'usinage 5 axes Mikron UCP 710, composé de trois axes linéaires et deux axes rotatifs. Cependant, elle peut être appliquée quelle que soit la structure du centre d'usinage 5 axes considéré. Deux briques logicielles élémentaires précédemment développées au LURPA en partenariat avec le L2S constituent le début du processus:

La première est le modèle non linéaire des axes du centre UCP 710 Mikron, qui a été initialement proposé par M. Susanu dans le cadre de sa thèse de doctorat au L2S de 2002 à 2005 [Susanu 2005], puis enrichi et validé par D. Prévost dans sa thèse de doctorat au LURPA de 2008 à 2011 [Prévost 2011a]. Le modèle a fait ses preuves pour simuler le comportement réel du centre d'usinage Mikron UCP 710 avec une grande fiabilité.

Le second support de cette thèse est un algorithme d'interpolation temporelle VPOp (Velocity Profile Optimization), développé par X. Beudaert lors de sa thèse de doctorat conduite au LURPA de 2010 à 2013 [Beudaert 2013], permettant de générer les consignes de position de référence, contenant un profil de vitesse d'avance optimisé, et respectant toutes les contraintes cinématiques des axes mentionnées auparavant.

Stratégie principale

La stratégie est illustrée dans la Fig. 3.

Dans une optique d'optimisation du processus, on peut noter que le réglage de l'entraînement des axes pourrait être différent en fonction de la surface usinée. L'idée est donc d'optimiser le contrôleur d'axe en fonction de la géométrie de la trajectoire d'usinage, avec pour objectif la réduction des erreurs de contour.

En particulier, au lieu d'utiliser des gains de commande fixes dans le contrôleur de position, les gains variables seront calculés tout en préservant la structure de commande en cascade classique avec anticipation.

L'idée de départ de cette thèse de doctorat est de rechercher une séquence optimale de gains de

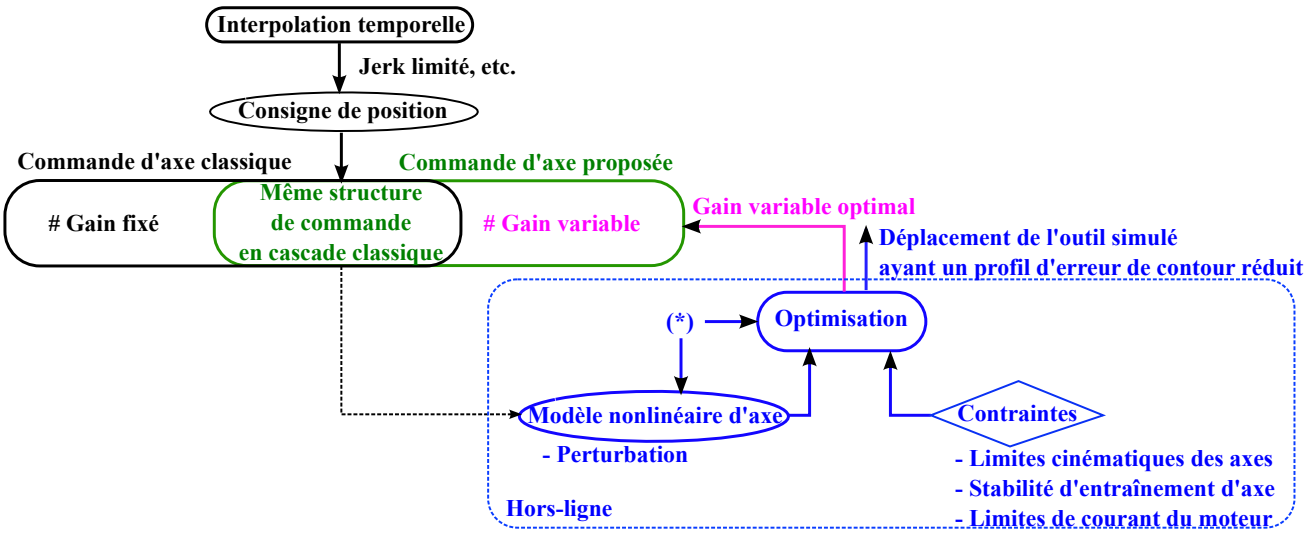

Figure 3: Vue globale de la stratégie proposée

commande variable, qui pourraient modifier de manière appropriée la dynamique d'entraînement des axes pour réduire l'erreur de contour pendant le processus d'usinage.

Les gains de commande sont générés hors ligne par résolution d'un problème d'optimisation et en utilisant le modèle non linéaire d'axes de la machine. De plus, le problème d'optimisation est soumis à des contraintes, dont les limites cinématiques des axes, le critère de stabilité des boucles d'asservissement et les limites de courant du moteur. En outre, l'erreur de contour est modélisée dans le cas de l'usinage 5 axes en bout et en flanc d'outil.

Une autre spécificité est l'utilisation de "l'horizon glissant", approche présente dans la commande prédictive (MPC), pour résoudre le problème d'optimisation pour l'ensemble de la trajectoire. Cette approche proposée, appelée "Offline Gain Adjustment (OGA)", permet de réduire l'erreur de contour pour l'usinage multi-axes grande vitesse, que ce soit en usinage 3 axes ou 5 axes.

Enfin, l'objectif est de développer les bases d'une méthode qui pourrait être mise en œuvre dans une CN commerciale.

Architecture du manuscrit

L'architecture du manuscrit est organisée comme suit :

Après l'introduction, le premier chapitre, "Géométrie et cinématique de trajectoire dans l'usinage à grande vitesse multi-axes", formule l'erreur de contour dans les cas d'usinage en bout et en flanc d'outil. Il met en évidence les contraintes cinématiques de la trajectoire de référence.

Tout d'abord, la chaîne d'usinage numérique est brièvement présentée. Les particularités de la génération de la trajectoire sont ensuite relevées.

Les techniques d'interpolation temporelle classiques sont expliquées et mènent à la présentation de la technique choisie pour générer la trajectoire de l'outil et les mouvements articulaires.

Le deuxième chapitre, "Commande de contour adaptative pour l'usinage multi-axes", consiste à développer l'approche proposée nommée "Off-line Gain Adjustment"(OGA).

Cette approche permet de pré-compenser l'erreur de contour définie au chapitre précédent, qui se produit dans le processus d'usinage. Tout d'abord, la structure classique de commande des axes est présentée. La caractéristique non linéaire du modèle d'axe est également détaillée. Deuxièmement, l'état de l'art relatif aux techniques avancées de commande d'axes est passé en revue. Ensuite, le cœur de ce chapitre consiste à formuler et à résoudre le problème d'optimisation sous contraintes lié à OGA.

Le but du troisième chapitre, "Simulations et discussions", est de mettre en évidence l'amélioration de la précision des contours grâce à l'approche OGA. Les effets du profil de gain variable optimal résultant, la vérification des contraintes et les effets de la configuration des paramètres d'OGA sont présentés et analysés à travers les résultats de simulation pour valider la pertinence de l'approche OGA.

Le quatrième chapitre, "Extensions OGA", consiste à explorer les applications d'OGA pour des objectifs diversifiés. Dans la première section, un gain global pour la commande en position est présenté et optimisé en utilisant l'approche OGA. Ensuite, la mise en œuvre d'OGA dans le cas de l'utilisation d'une structure de commande prédictive généralisée est effectuée. Les résultats de simulation correspondants prouvent les possibilités offertes par OGA dans la pré-compensation de l'erreur de contour au sein de différents types de contrôleurs de position.

Enfin, les "Conclusions et perspectives" visent à évaluer non seulement les contributions scientifiques mais aussi les perspectives de ces travaux.

Contributions de la thèse

Les principales contributions de cette thèse sont présentées ci-dessous.

La première contribution porte sur la formulation de l'erreur de contour en fraisage 5 axes en bout d'outil et en flanc d'outil. Pour obtenir de telles formulations d'erreur de contour, la géométrie de la trajectoire FAO est détaillée et certaines hypothèses sont proposées. En outre, la cinématique prise en compte lors de l'étape d'interpolation temporelle est également exploitée.

Une loi de commande d'axe adaptative est proposée fondée sur l'approche Offline Gain Adjustment (OGA) pour compenser l'erreur de contour pendant l'usinage. Le diagramme général d'OGA est illustré dans la Fig. 4.

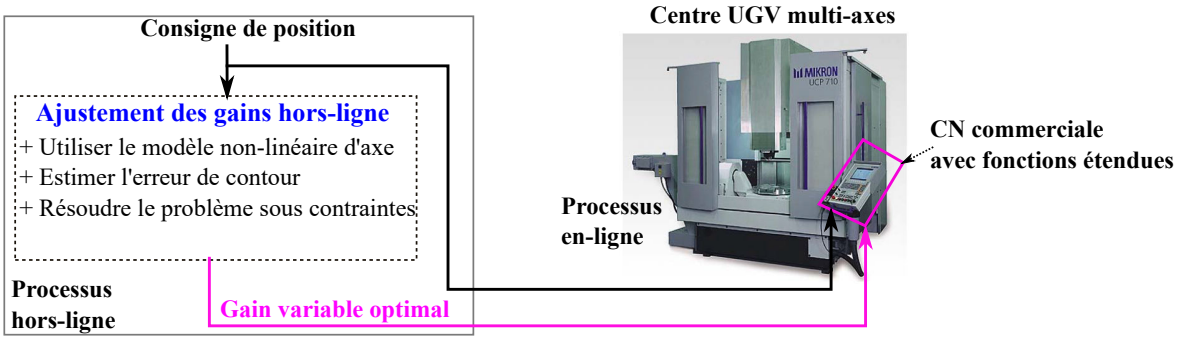

Figure 4: Idée générale d'OGA

L'une des contributions importantes de cette thèse est le développement des deux techniques de résolution du problème d'optimisation. Elles sont fondées sur le principe du réglage des gains à l'intérieur d'une fenêtre glissante sur la trajectoire.

Dans la première technique de résolution, les fenêtres considérées sur la trajectoire sont calculées en tenant compte des caractéristiques de courbure du profil. Ensuite, le gain à l'intérieur de chaque fenêtre est optimisé en se fondant sur une fonction d'évolution de gain prédéfinie. Une caractéristique de cette première technique est qu'il n'y a pas de chevauchement des fenêtres sur la trajectoire considérée.

En revanche, la deuxième technique de résolution prédit les gains constants optimaux sur un horizon glissant, mais seul le premier gain optimal est conservé et l'horizon est avancé d'un pas en avant. Cette technique de résolution nécessite une quantité de calcul plus importante

que la première, mais elle permet d'obtenir un profil de gain de commande plus flexible. Par conséquent, cette seconde technique de résolution permet une réduction de l'erreur de contour plus importante que la première méthode.

Enfin, une nouvelle idée autour d'un gain de commande global est proposée. Ce gain peut être intégré dans les différents types de contrôleurs, classiques ou avancés. Cette méthode peut garantir non seulement l'amélioration de la précision de l'usinage, mais aussi un temps de calcul raisonnable. Cela élargit les possibilités de l'approche OGA proposée dans un environnement industriel.

Les deux techniques de résolution étant différentes dans leurs propres principes, leurs configurations sont également définies différemment. Les impacts des configurations des paramètres sur les performances d'OGA sont discutés. Le choix du nombre maximal de gains potentiels, du facteur de variation de gain et des longueurs d'horizon se sont avérés importants pour la performance d'OGA. De plus, l'effet du facteur de pondération sur la gestion du compromis entre le lissage du profil de gain et la précision du contournage est également étudié.

Les résultats de simulation fondés sur un simulateur d'usinage validé ont montré que le niveau de précision du suivi de contour et le temps de calcul sont différents selon les différentes techniques de résolution et le nombre d'axes et les gains de commande impliqués dans l'optimisation d'OGA. Les résultats ont également vérifié que les réponses de l'OGA ont respecté toutes les contraintes concernées. Les relations entre les gains variables obtenus, la réponse en jerk, la vitesse d'avance résultante et les profils d'erreur de contour ont été mis en évidence. Grâce à la fiabilité du simulateur de machine, on peut dire que l'erreur de contour est considérablement réduite grâce à l'approche proposée.

De plus, une interface OGA-GUI est construite pour faciliter son utilisation et faciliter les applications pratiques.

Perspectives

Une perspective est de prendre en compte d'autres effets non linéaires, ainsi que des perturbations, comme par exemple les flexibilités de la structure machine, ou encore les efforts de coupe.

Par ailleurs, d'autres techniques d'optimisation, telles que méthode heuristique ou algorithme génétique, devraient être investiguées. L'objectif est de réduire la charge de calcul pour résoudre le problème d'optimisation et de produire de meilleurs gains de commande optimaux pour la réduction des erreurs de contour.

Une autre perspective est l'analyse des variations de gains obtenus pour construire en amont des motifs de variation en fonction des caractéristiques géométriques du profil à usiner. Cela pourrait faciliter l'obtention de la variation des gains sur toute la trajectoire.

Enfin et surtout, des essais expérimentaux d'OGA devraient être réalisés sur une machine outil possédant une CN ouverte.

Introduction

Context

The advanced machining techniques are always the backbone of the manufacturing industries. One of such techniques, High Speed Machining (HSM), is the main subject of this PhD thesis. It replaces the Conventional Speed Machining (CSM) in sculpturing intricate parts in aerospace, automotive, die or mold industries, etc. This is because high speed machining increases material removal rates and enhances surface finish. As a result, an economical benefit can evidently be obtained.

Generally, high speed machining involves many processes, as shown in Fig. 5, from Computer Aided Design (CAD); trajectory generation by Computer Aided Manufacturing (CAM); execution treatments by Computer Numerical Control (CNC), including feedrate interpolation and axes control; to structural behaviors inside Machine Tool (MT). Finally, the tool displacement is obtained, as well as feedrate and Contour Error (CE) profiles along the machining trajectory.

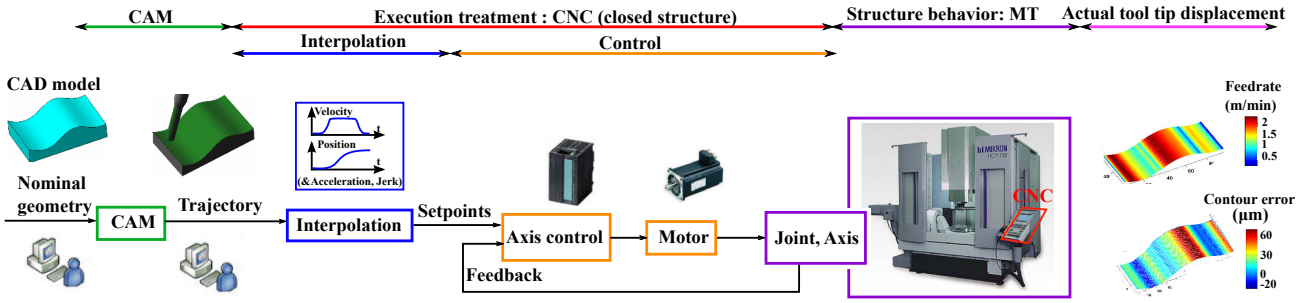

Figure 5: Simplified high speed machining procedure

Motivation and framework

Due to the high competition in the manufacturing industries, there are increasing demands for implementation of high speed machining strategies. To boost the performance of high speed machining, this PhD thesis mainly focuses on the improvement of the axis control stage, particularly the position controller of axis drive in the CNC process. The general motivation is that if the control action performs really well, the part finishing will be very close to the desired one. Consequently, the hand finishing and polishing steps after machining can be eliminated or at

least will require less effort. Hence, the cutting quality, productivity, and economical benefit can obviously be achieved.

The framework of the current research is highlighted within a multi-input and multi-output process of machining in Fig. 6.

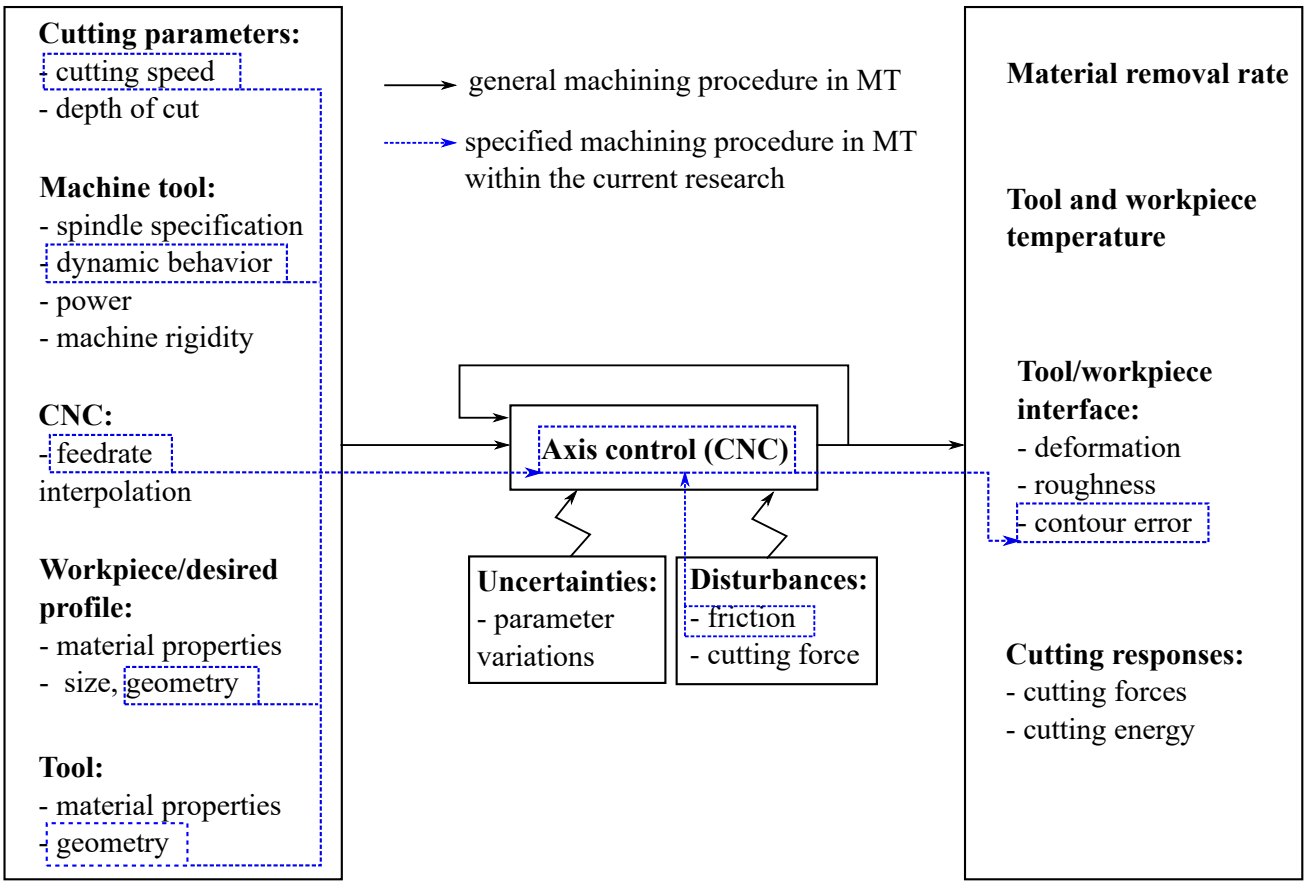

Figure 6: Machining process on machine tool [Dagiloke 1995] and specified concerns in the research framework

In this process, the cutting parameters (e.g. cutting speeds, depth of cut, etc), configuration and specification on machine tool (e.g. spindle specification, dynamic behavior, etc), CNC (feedrate interpolation, axis control), workpiece (e.g. material properties, size, geometry, etc) and tool (e.g. material properties, geometry, etc) are designated as the inputs. Meanwhile, the output effects are such as material removal rate, tool and workpiece temperatures, deformation of machined part, cutting forces, etc. The core part of the machining process on machine tool is the axis control realized in a feedback manner, being subject to uncertainties (e.g. variation

of drive parameters), and disturbances (e.g. friction, cutting force). In the limit of the current research, five input factors are mainly considered, including cutting speeds, dynamic behavior of machine tool (i.e. the axes), feedrate, geometry of desired part/surface and tool. Only friction is considered as the disturbance on axis control. At output, only contour error profile is taken into consideration.

To enhance the performance of the industrial CNC, interpolation techniques, computer processor, motor and actuator have been much improved. However, its control structure and strategy have undergone less improvement. In fact, each axis drive is controlled by CNC through a classical cascaded structure, including position, velocity and current loops ranging from outer to inner loops respectively. While the two inner loops are controlled by Proportional - Integral (PI) controllers, the external position loop is controlled by a Proportional (P) controller, combined with the Feed Forward (FFW) action for the velocity loop [Altintas 2000a].

During control process, it is known that small discontinuities in the trajectory can excite the natural modes of the mechanical structure, causing machine vibration. Therefore, the reference trajectory firstly needs to be smoothed. To do this, the trajectory planing is performed by considering the kinematic limitations of the machine, especially the limited jerk [Erkorkmaz 2001a, Barre 2005, Beudaert 2012]. Afterwards, the controller is expected to produce the resulting trajectory respecting all of the kinematic constraints of the reference one.

Considering that machining free-form surface is primarily concerned in this study, there are many reasons saying that with the above classical axis control, it is difficult to obtain high contouring accuracy. On the one hand, in high speed machining, the high cutting speed leads to high feedrate. By interpolating such a feedrate, the required axis velocity, acceleration and jerk are also increased. Thus, by following the free-form profile the machine tool axes may have high kinematic characteristics and high frequency. On the other hand, the classical axis control structure has a limited bandwidth. Furthermore, machine tool contains uncertainty characteristics and disturbances, such as variation in drive parameters due to the varying workpiece mass and lubrication condition, coulomb friction and cutting force [Altintas 2000a, Altintas 2000b, Erkorkmaz 2001b, Tang 2013, Khoshdarregi 2014]. These detrimental facts increase tracking error, and generate large contour error, which is classically defined as the orthogonal distance between the actual tool contact point and the desired profile.

To cope with the above challenges, researchers proposed many advanced and intelligent con-

trol strategies as an alternative for the classical one, such as Cross Couple Control (CCC) technique [Koren 1980, Koren 1991, Zhao 2013], Sliding Mode Control [Altintas 2000b], Adaptive Robust Control (ARC) [Yao 1997, Davis 2015], Fuzzy Logic Control [Ngo 2013], Neural Network [Lin 2006], Model Predictive Control (MPC) [Susanu 2006, Tang 2012], etc. These approaches are more or less validated by simulation results and/or experimental tests using an open CNC [Altintas 1994, Erkorkmaz 2001b, Beudaert 2014]. However, the implementation of the advanced controllers into the commercial version of CNC is not achieved yet. This is due to the fact that the CNC up to now permits only a limited access to the core of the controllers due to their close features. Another reason is that almost all the proposed control strategies require the modification of the classical control structure. That may be inconvenient and very expensive for the CNC manufacturer. Therefore, the previously mentioned classical control structure is still in use in the commercial CNC.

In summary, this PhD thesis "Variable gain contouring control for multi-axis machine tools" aims at improving the contouring accuracy in multi-axis high speed machining of free-form surfaces, by exploiting all possibilities of the classical control structure of axis drive, proposing solutions that the manufacturers can implement into their commercial CNC. The enhancement of contour following means that the contour error should be controlled to be as small as possible. In this research, except for the friction, other sources of disturbance and uncertainties are neglected. Moreover, considering machine vibration, the feedrate interpolation step produces the axis motions, respecting axis kinematic constraints, especially the limited jerk. This thesis refers to a method to generate optimal variable control gains for pre-compensating the contour error in machining process, using optimization techniques. The proposed optimization is here subject to the constraints, including the axis kinematic (velocity, acceleration and jerk) limitations, the stability of the axis drive and the motor current limits. Other factors causing the machine vibration are not taken into consideration.

This PhD work is carried out at the University of Paris-Saclay, France, within the cooperation between two French research laboratories: Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec and Laboratoire Universitaire de Recherche en Production Automatisée (LURPA), ENS Paris-Saclay. This work is registered under N° 2014-812D – Projet OMEGA and supported by DIGITEO foundation. The proposed method is specifically built for the 5-axis machining center Mikron UCP 710, consisting of three linear and two rotary axes. However, it is expected to be applicable to the structures of 5-axis machining centers. There are two starting points favoring

this work.

The first one is the nonlinear model of the translation axes of the Mikron UCP 710, that has been proposed by M. Susanu within her PhD thesis at L2S from 2002 to 2005 [Susanu 2005], enhanced and validated by D. Prévost in his PhD thesis at LURPA from 2008 to 2011 [Prévost 2011a]. The model was proven to simulate the real behavior of the Mikron UCP 710 machining center with high confidence.

The second support of this thesis is a velocity optimization profile (VPOp) algorithm, developed by X. Beudaert during his PhD thesis conducted at LURPA from 2010 to 2013 [Beudaert 2013], allowing the generation of reference position setpoints, containing an optimized velocity profile, and respecting all the axis kinematic constraints.

Strategy

The main strategy of this PhD thesis is illustrated in Fig. 7.

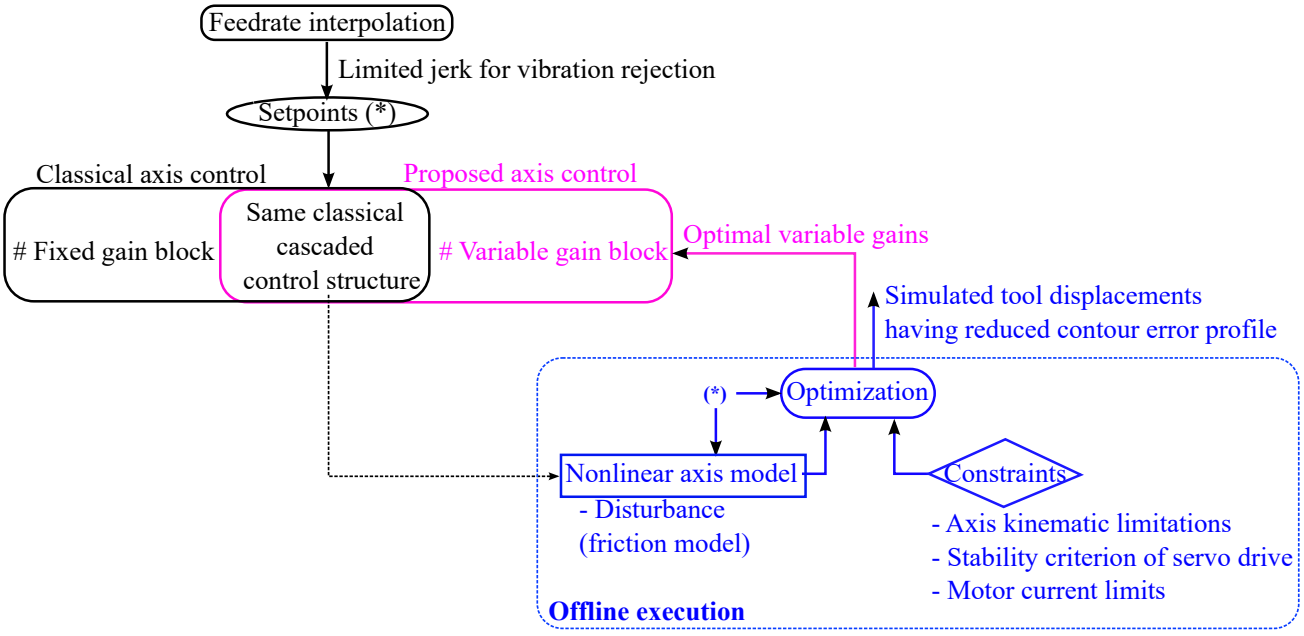

Figure 7: Global view of this PhD thesis

It is observed by CNC users that axis drive tuning can be different in function of the machined

surface. Thus, the idea is to optimize the axis controller depending on the geometry of the machining trajectory, with the objective of contour error reduction.

Particularly, instead of using fixed proportional and feed forward control gains in the position controller, variable values of such gains will be computed while preserving the classical cascaded control structure with feed forward.

The original idea of this PhD thesis is to seek an optimal set of variable control gains that could excite appropriately the axis drive dynamic to reduce the contour error during machining process.

Control gains are generated off-line by solving an optimization problem and using the nonlinear axis model of the machine. Moreover, the optimization problem is subject to constraints, including axis kinematic limitations, stability criterion of servo drives and motor current limits. Moreover, the contour error is considered in both 5-axis point milling and 5-axis flank milling.

The off-line execution is based on the nonlinear axis model, in which the disturbance coming from friction model is considered.

It is well known that when the friction is concerned, the feed forward friction compensation is widely used in CNC system to overcome the detrimental effects of friction at motion reversals, especially for contouring accuracy. However, the current nonlinear axis model, served as the fundamental base of this thesis, has not been equipped yet such a compensation technique. Therefore, the control law for contour error reduction is proposed in this thesis without regard to this situation as well. It is assumed that the feed forward friction compensation is well designed in CNC and the proposed control law serves as an advanced supplementary approach to improve more the contouring accuracy in CNC machining.

Another specificity is the use of "receding horizon" approach of Model Predictive Control (MPC) to solve the optimization problem for the whole trajectory. This proposed approach, so-called Off-line Gain Adjustment (OGA), can be used to reduce the contour error for multi-axis high speed machining, either 3-axis or 5-axis machining.

Ultimately, the objective is to build the foundations of a method that could be considered for an implementation in a commercial CNC.

Thesis organization

The manuscript architecture is summarized as follows:

After the introduction, the first chapter, "Tool path geometry and kinematics in multi-axis high speed machining", is to formulate the contour error in multi-axis high speed machining and to highlight the kinematic constraints of the reference trajectory.

Firstly, the digital chain of machining is briefly introduced. The specifics in the generation of the programmed tool path is then discussed. After that, the contour error formulations for either point or flank milling are defined.

It is followed by the discussions of the feedrate interpolation techniques leading to presentation of the chosen technique to generate the tool trajectory and the joint motions.

The second chapter, "Adaptive contouring control in multi-axis high speed machining", consists in developing the proposed approach called "Off-line Gain Adjustment" (OGA).

This approach allows to pre-compensate the contour error defined in chapter two, that occurs in the machining process. Firstly, the introduction of the classical axis control structure is presented. The nonlinear feature of the axis model is also provided. Secondly, the state of the art related to the advanced techniques of contouring control is reviewed.

Then, the core section of this chapter is to formulate and solve the optimization problem under constraints related to OGA.

The intent of the third chapter, "Simulations and discussions", is to highlight the improvement of contouring accuracy thanks to the OGA approach proposed in chapter three. Impacts of the resulting optimal variable gain profile, constraint verification and effects of OGA's parameters configuration are shown and discussed through simulation results to validate the performance of the OGA approach.

The fourth chapter, "OGA extensions", is to explore the applications of OGA for diversified purposes. In the first section, the global gain of the position controller is introduced and then tuned by using the OGA approach. Next, the application of OGA in the case of using a Generalized Predictive Control structure is discussed. The corresponding simulation results prove the valuable possibilities of OGA in pre-compensating the contour error within different kinds

of position controllers in multi-axis high speed machining.

Finally, "Conclusions and perspectives" aims at evaluating not only the scientific contributions but also the perspectives of this PhD thesis.

Chapter 1

Tool path geometry and kinematics in multi-axis high speed machining

The main goal of this chapter is to highlight tool path geometry and kinematic execution in multi-axis high speed machining, in order to support the formulation of axis control problem in the next chapter.

Firstly, the digital chain of machining is introduced. Multi-axis tool path computations, referring to tool geometry, tool positioning, geometrical errors, tool path description and milling context, are represented afterwards.

In order to propose a method to pre-compensate contour error in off-line process, this error must be estimated based on a machine simulator. Thus, the next section is dedicated to contour error estimation approach.

Lastly, relating to kinematic execution of trajectory, feedrate interpolation is carried out to evaluate its effects on the axis control and machining process.

1 Introduction of machining process

The digital chain of machining is shown in Fig. 1.1. It can be classified into five major stages, as follows:

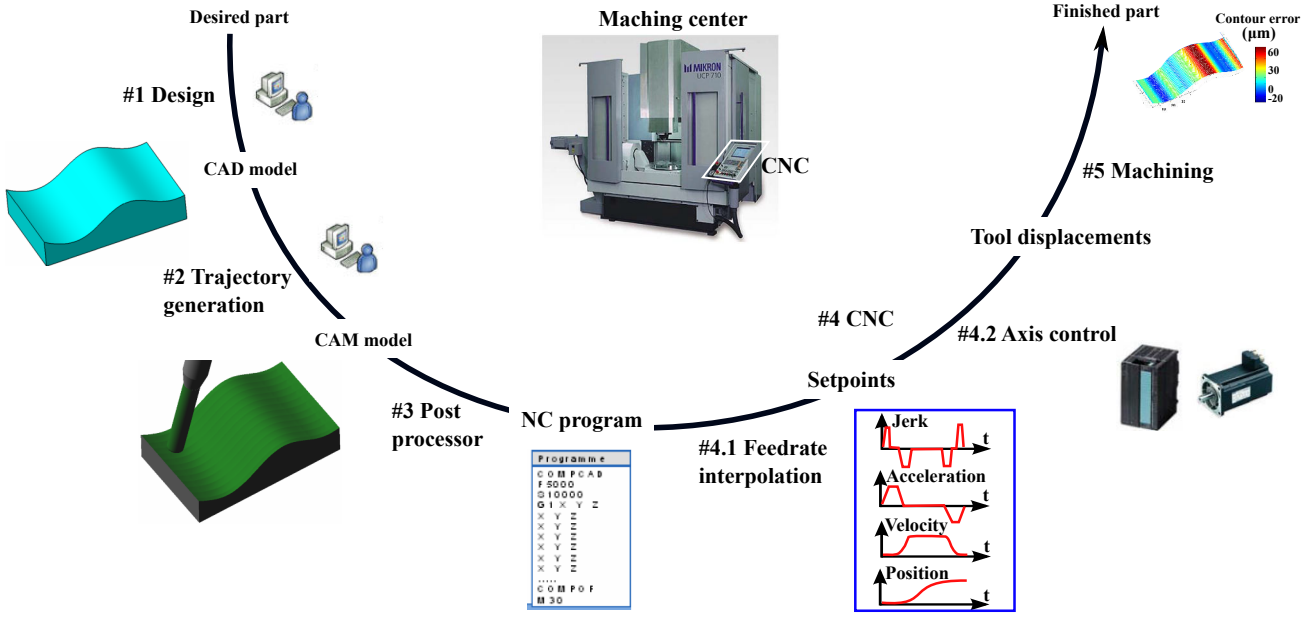

Figure 1.1: Digital chain of machining

#1 Design: This stage allows to obtain a nominal surface of the desired part.

#2 Trajectory generation: From the nominal surface, CAM computes a set of reference contact points between the tool and the workpiece. Based on these points, a programmed tool path is determined. Most of time, such a tool path is represented by a set of discrete tool location points and tool orientation vectors. These data constitute the CL data, also called CL file.

#3 Post processor [Jung 2002, She 2007]: This is dedicated to transform the programmed tool path into the adapted language for numerical control (NC) in CNC, so-called "NC program".

#4 CNC [Schmitz 2001, Omirou 2005, Mori 2013, Yuen 2013, Beudaert 2013, Besset 2017]: This stage consists of two tasks: feedrate interpolation and axis control. The former allows to generate a trajectory having a feedrate profile, in such a way that the corresponding axis motions, which are obtained by solving the Inverse Kinematic Transformation (IKT), respect the machine constraints, e.g. the axes velocity, acceleration, and jerk limitations, to limit the machine vibration, etc. The later controls the axes motions through closed control loops, axes drive and motor. The combined motions of all axes allow to obtain the tool displacements.

#5 Machining [Neugebauer 2007]: This task is performed in machining center. It is in fact the relative movements between the tool and the workpiece, controlled by the axis control in CNC, to sculpture the machined part.

2 Multi-axis tool path computation

The aim of this section is to analyze the way that tool path is programmed in the context of multi-axis machining.

2.1 Tool geometry

There are many types of tools with different geometries. However, nearly all tools used in milling community can be defined from one generic form, called APT parameterization [APT 1983], as seen in Fig. 1.2.

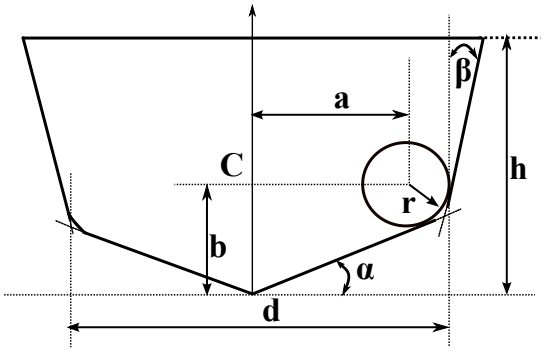

Figure 1.2: APT parameterization

From APT parameterization, the general parameters a, b, d, h, α, β and r are modified and adapted to form different tool types. For example, the three most commonly used tool types: hemispheric tool (ball nose or ball end cutter), toric tool (bull nose end mill cutter) and cylindrical tool, are illustrated in Fig. 1.3 a, b, and c, respectively.

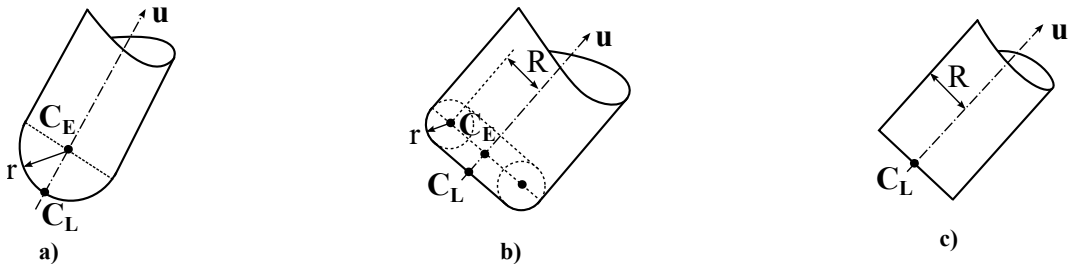

Figure 1.3: Tool types: a) Hemispheric tool; b) Toric tool; c) Cylindrical tool

The main geometrical characteristics of the above tools are given as follows:

- $\mathbf{C_E}$: Center point of the tool,
- $\mathbf{C_L}$: Cutter location described in the NC program, which is generally the extremity point of the tool,
- $\mathbf{u}$: Unit vector of the tool axis orientation,
- r: Radius of the hemispheric tool and small radius of the toric tool,
- R: Radius of the cylindrical tool and large radius of the toric tool.

2.2 Tool positioning

The programmed tool path is characterized by a couple of values $(\mathbf{C_L}, \mathbf{u})$, that represents the tool positioning.

To calculate the above components of the programmed tool path, CAM firstly computes the tool contact points, $\mathbf{C_C}$, onto the nominal surface. Using different tools for one specific milling technique, for example point milling illustrated in Fig. 1.4, leads to the fact that the tool location points are formulated differently from (1.1) to (1.4) [Duc 1998].

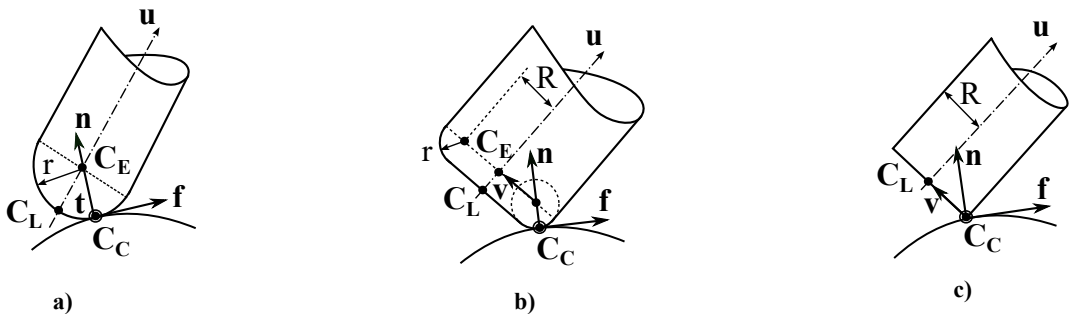

Figure 1.4: Point milling with a) hemispheric tool; b) toric tool; c) cylindrical tool

- for hemispheric tool:

$$\begin{aligned} \mathbf{C_E} &= \mathbf{C_C} + r.\mathbf{n} \\ \mathbf{C_L} &= \mathbf{C_C} + r.\,(\mathbf{n} - \mathbf{u}) \end{aligned} \tag{1.1}$$

- for toric tool:

$$\begin{aligned} \mathbf{C_E} &= \mathbf{C_C} + r.\mathbf{n} + R.\mathbf{v} \\ \mathbf{C_L} &= \mathbf{C_C} + r.\,(\mathbf{n} - \mathbf{u}) + R.\mathbf{v} \end{aligned} \tag{1.2}$$

  with

$$\mathbf{v} = \frac{\mathbf{u} \wedge \mathbf{n}}{\|\mathbf{u} \wedge \mathbf{n}\|} \wedge \mathbf{u} \tag{1.3}$$

- for cylindrical tool:

$$\mathbf{C_L} = \mathbf{C_C} + R.\mathbf{v} \tag{1.4}$$

  with $\mathbf{v}$ defined in (1.3).

in which $\mathbf{f}$, $\mathbf{n}$ and $\mathbf{t} = \mathbf{f} \wedge \mathbf{n}$, are the unit feed, surface normal and transverse vectors, respectively.

The tool axis direction $\mathbf{u}$ is defined by tilt angle θ_t and yaw angle θ_n in CAM, as seen in Fig. 1.5. In fact, $\mathbf{u}$ is obtained by two consecutive rotations: firstly a rotation of $\mathbf{n}$ by an angle of θ_t around $\mathbf{t}$ and then the obtained vector is rotated by an angle of θ_n around $\mathbf{n}$ [Lauwers 2003].

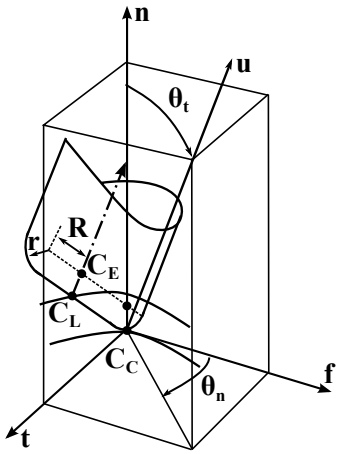

Figure 1.5: Tool positioning parameters

2.3 Geometrical errors

In the tool path planning, the tool movement involves the longitudinal and transversal tool steps, as illustrated in Fig. 1.6.

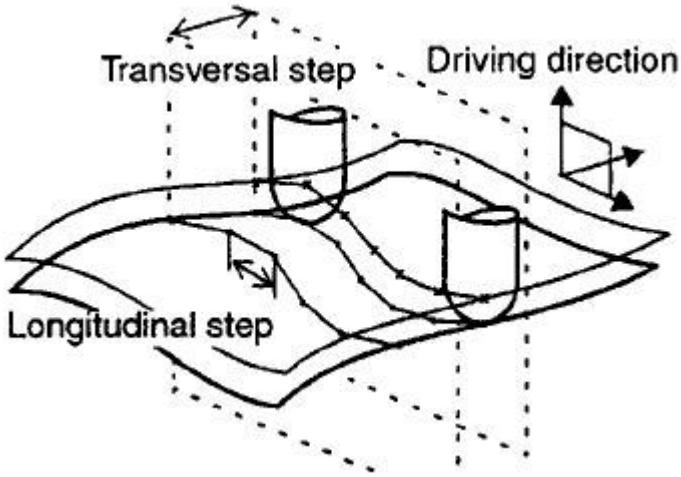

Figure 1.6: The tool path planning [Duc 1999]

The tool contact points $\mathbf{C_C}$ are determined in CAM, in a way that the tool movement envelope, illustrated in Fig. 1.7, respects the admissible geometrical errors, which are usually chordal error and scallop height.

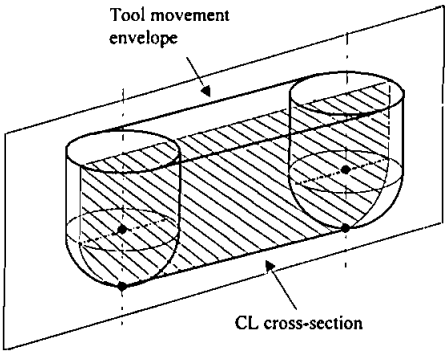

Figure 1.7: Tool movement envelope and $\mathbf{C_L}$ cross-section [Kim 1995]

The chordal error e, illustrated in Fig. 1.8, represents the maximum deviation between the nominal surface and the tool movement envelope generated by the interpolation of the tool positioning along a longitudinal tool step [Kim 1995].

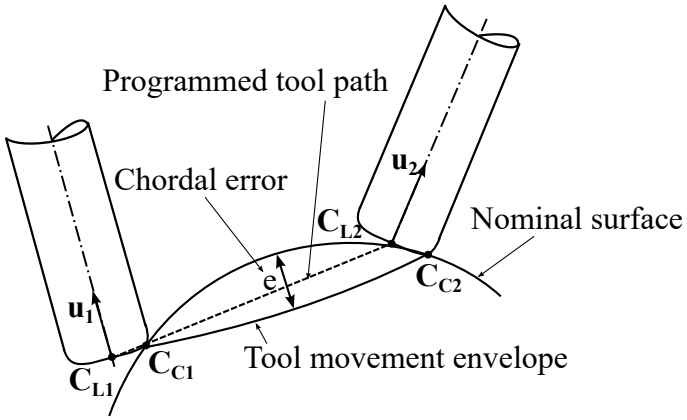

Figure 1.8: Chordal error

The scallop height, h, represents the remaining material after a transversal tool step. One example for machining a flat surface by a hemispheric tool is illustrated in Fig. 1.9.

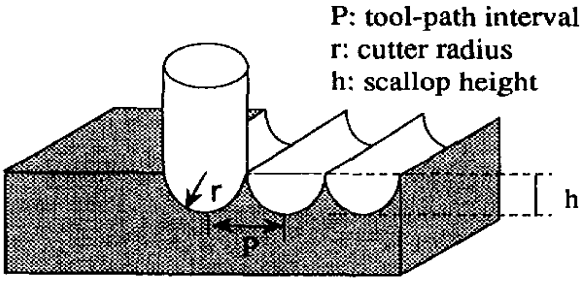

Figure 1.9: Scallop height of a flat surface machined by a hemispheric tool [Lin 1996]

By respecting the conditions of chordal error $e \leq e_{max}$ and of scallop height $h \leq h_{max}$, the tool contact points $\mathbf{C_C}$ are generated by CAM.

2.4 Tool path description

In multi-axis machining, there are two main formats for the description of tool path: either by linear segments (linear interpolation - G1) or by curves (polynomial interpolation - Bspline) [Langeron 2004]. The former, which is more commonly used in the industrial community than the latter, is presented below.

Linear interpolation allows to describe tool path through a series of tool positionings. The tool location points are under the form of linear segments ($\mathbf{C_L}$ point - ISO 3592), while tool axis orientations are interpolated in workpiece frame, as illustrated in Fig. 1.10. Each tool positioning constitutes a block of machining program, including three position components (x, y, z), and three orientation components (i, j, k).

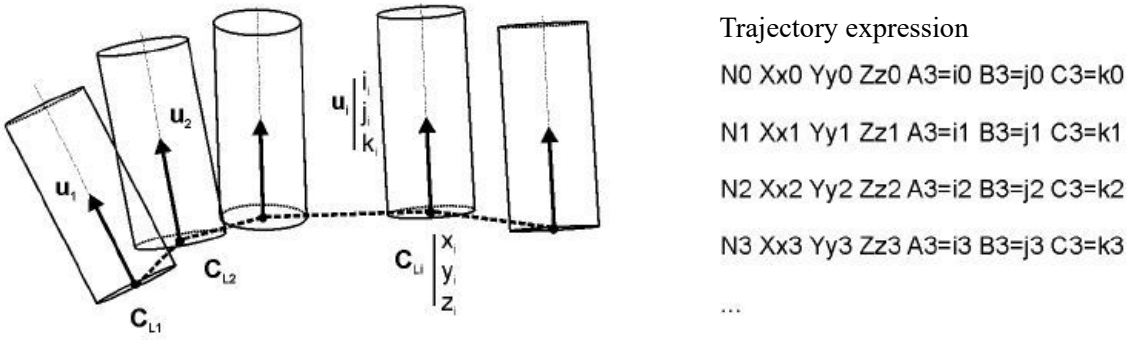

Figure 1.10: Tool path description by linear interpolation [Lavernhe 2006]

In principle, when the tool moves in the piecewise linear way, the tool contact point $\mathbf{C_C}$ follows also the piecewise linear way, which approximates the nominal surface within an allowable chordal error. However, this is only true for 3-axis machining. As discussed in [Bohez 2002], in 5-axis machining, even if the linear interpolation is used in CAM, the real $\mathbf{C_C}$ path is not formed by linear segments, but a curve, as seen in Fig. 1.11. This fact is due to the effect of tool orientation. Thus, in 5-axis machining, the calculation of the chordal error is more complex than in 3-axis machining, because it should take the deviation generated by the tool orientation into account.

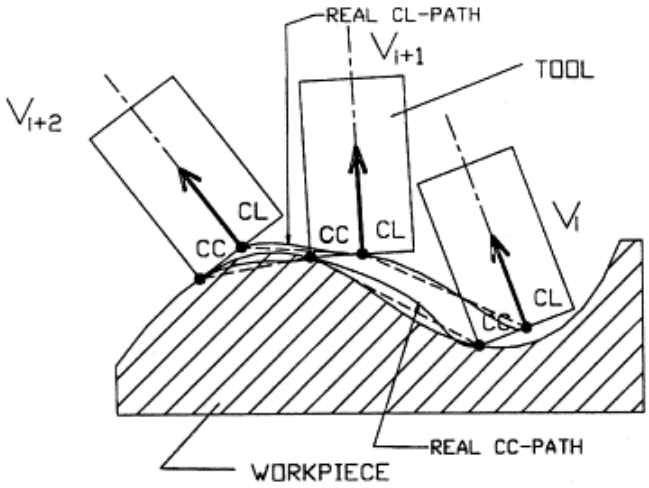

Figure 1.11: Convex/concave part - $\mathbf{C_C}$ and $\mathbf{C_L}$ points [Bohez 2002]

2.5 Milling contexts

Another factor that affects the configuration of the programmed tool path is the milling context, referring to point milling, flank milling, and combined milling, as illustrated in Fig. 1.12.

In point milling, the tool is in contact with the workpiece through a contact point $\mathbf{C_C}$. In flank milling, the tool intersects the workpiece through a contact segment on the tool trunk. The last milling context combines the above two kinds of intersection between the tool and the workpiece.

Fig. 1.12 also highlights differences between the 3-axis and 5-axis configurations. The former refers to a machine having three linear axes, in which the tool orientation is fixed in vertical or horizontal direction depending on machine structure. The latter differs from its counterpart due to the addition of two rotary axes. Thus, the tool orientation can be changed relatively to the workpiece during machining process.

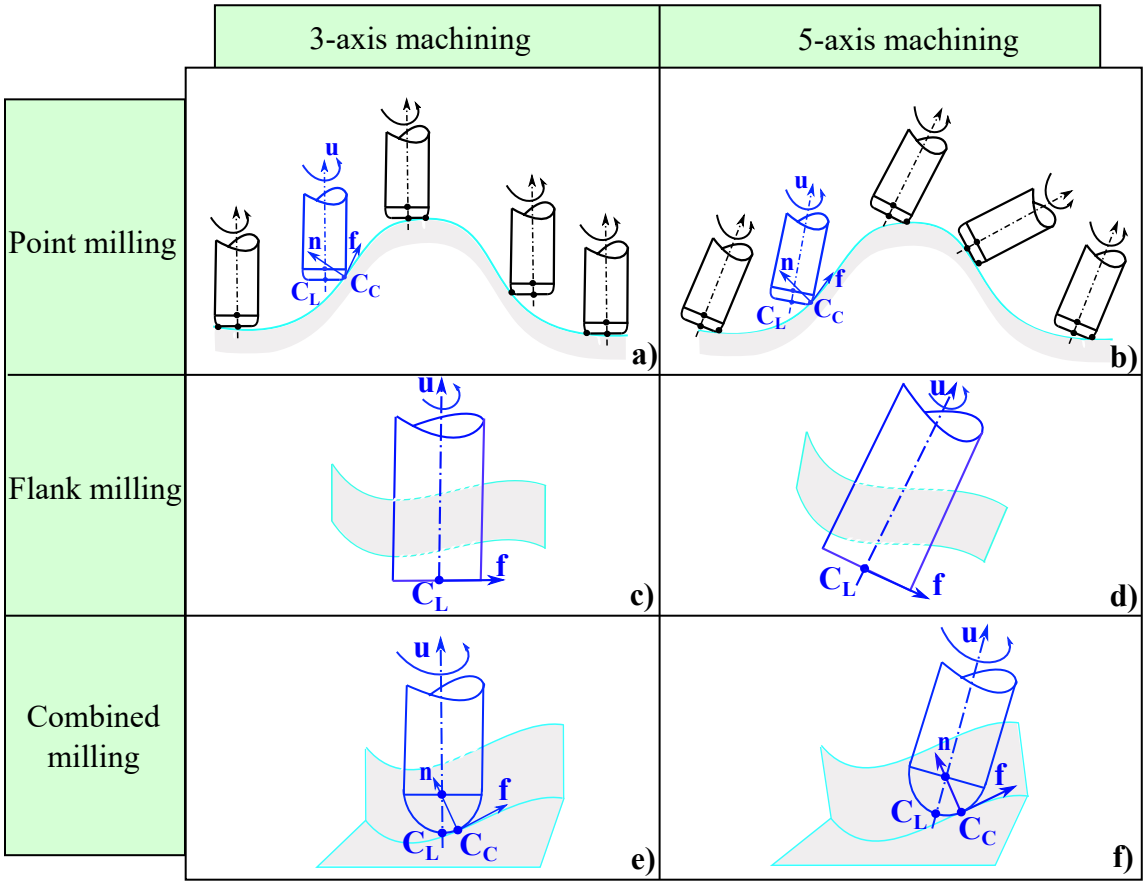

Figure 1.12: Multi-axis machining contexts

2.6 Conclusion

According to the above discussions related to the multi-axis tool path computation, there are important remarks as follows:

- Different tool geometries lead to different formulas of the tool positioning.
- It is important to note that the tool does not remove all the material along its programmed tool path, because of the inherent existence of the chordal error and scallop height.
- The 5-axis machining center makes the milling more flexible than the 3-axis one, due to a variable tool orientation. However, it also places the challenges in computing and respecting the maximum chordal error and scallop height.
- The different milling contexts have their own characteristics in sculpturing the workpiece,

especially the different contacts between the tool and the workpiece, that should be considered at the axis control stage.

3 Contour error estimation in multi-axis machining

Following, contour error formulation will be developed for 5-axis machining. This can obviously be applied for 3-axis machining as well. For simplicity, the superscripts m and w are denoted for the articular (machine axis) and workpiece spaces, respectively. The trajectory defined by the NC program is considered as the reference, while the trajectory generated by the effective tool movement during machining is considered as the actual one. The superscripts r and a are therefore used for the reference and actual parameters.

3.1 Contour error estimation in 5-axis point milling

3.1.1 Problem formulation

The definition of contour error in 5-axis point milling is explained through the schematic procedure illustrated in Fig. 1.13.

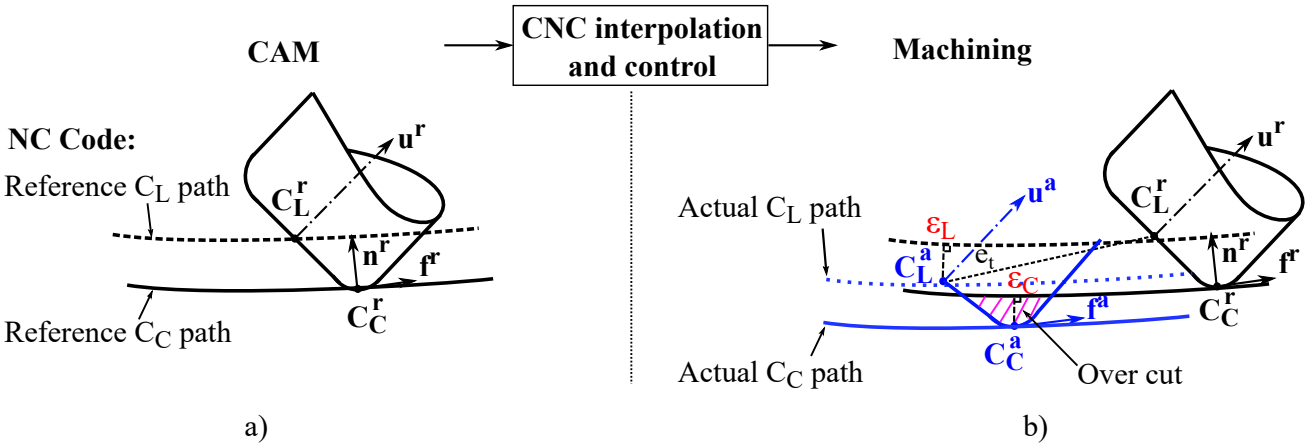

Figure 1.13: a) Reference tool positioning; b) Actual tool positioning with contour error

As it can be seen in Fig. 1.13.a, CAM generates a programmed tool path, characterized by a set of $(\mathbf{C_L^r}, \mathbf{u^r})$, in which the reference $\mathbf{C_L}$ path can be represented by a series of reference tool

location points $\mathbf{C_L^r}$ and tool axis orientation $\mathbf{u^r}$. If the tool follows exactly such a programmed tool path, it will sculpture the workpiece to obtain a desired machining profile, that approximates the nominal surface with the allowable geometrical errors, as discussed in § 2.3. This desired machining profile can be seen as the reference $\mathbf{C_C}$ path, represented by a set of reference tool contact points $\mathbf{C_C^r}$.

The above programmed tool path is interpreted into NC code, that is then fed into the CNC. After that, tool axes are controlled to follow the axis position setpoints, that are generated from the programmed tool path through feedrate interpolation and the IKT, without knowledge of the reference tool contact points, but only the couple $(\mathbf{C_L^r}, \mathbf{u^r})$.

In fact, due to the limited bandwidth of the axis drives and the nonlinear characteristics, such as friction or cutting force, the actual tool positioning $(\mathbf{C_L^a}, \mathbf{u^a})$ lags behind the reference one $(\mathbf{C_L^r}, \mathbf{u^r})$, as illustrated in Fig. 1.13.b. Such lag distance is defined as the tool tracking error, e_t, in (1.5).

$$e_t = \|\mathbf{C_L^r} - \mathbf{C_L^a}\| \tag{1.5}$$

As it can be seen, there is a deviation between the actual tool path and the programmed one. Sets of the actual tool location points $\mathbf{C_L^a}$ and of the actual tool contact points $\mathbf{C_C^a}$ represent the actual $\mathbf{C_L}$ and $\mathbf{C_C}$ paths, respectively.

In 3-axis machining, many works in the literature [Koren 1983, Erkorkmaz 2001b, Yao 2012] defined contour error in point milling as the orthogonal distance ($distance_\perp$) between the actual tool location $\mathbf{C_L^a}$ and the reference $\mathbf{C_L}$ path, denoted as ε_L and expressed in (1.6).

$$\varepsilon_L = distance_\perp\left(\mathbf{C_L^a}, \text{Reference } \mathbf{C_L} \text{ path}\right) \tag{1.6}$$

The purpose of such a contour error definition is to approximate the effective cutting error, caused by the actual tool contact point $\mathbf{C_C^a}$. This is because the information of the tool contact points is not available in the axis control and machining stages. This error can be represented by the orthogonal distance between the actual tool contact point $\mathbf{C_C^a}$ and the reference $\mathbf{C_C}$ path, denoted as ε_C, and expressed in (1.7).

$$\varepsilon_C = distance_\perp\left(\mathbf{C_C^a}, \text{Reference } \mathbf{C_C} \text{ path}\right) \tag{1.7}$$

In 5-axis machining, to improve the approximation accuracy of the effective cutting error ε_C, [Altintas 2010, Yang 2015a] considered a complementary error related to the deviation of tool

orientation. As it can be seen in Fig. 1.14, even though the tool location point can be tracked with high accuracy, the tool orientation deviation can still cause the cutting error ε_C. The error related to the deviation of the tool orientation is denoted as ε_{u} and defined in (1.8).

$$\varepsilon_{\mathrm{u}} = angle\,(\mathbf{u^r}, \mathbf{u^a}) \tag{1.8}$$

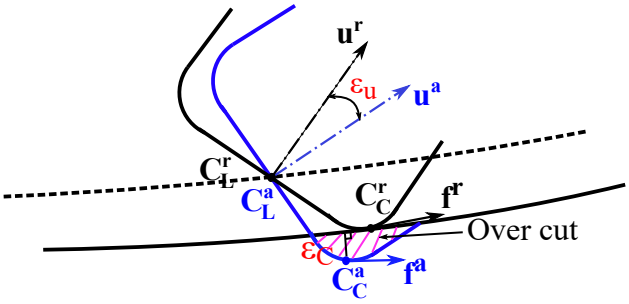

Figure 1.14: Cutting error caused by tool orientation deviation

According to the above discussions, it can be noticed that the main objective of contour error definition, ε_{L} or $(\varepsilon_{\mathrm{L}}, \varepsilon_{\mathrm{u}})$, is to estimate the effective cutting error, ε_{C}. Main approaches for contour error determination are briefly presented in next paragraph.

3.1.2 Literature review

The strategies for contour error estimation have some specificities as follows.

Firstly, they depend on the definition of the contour error. Whether contour error is defined by ε_{L} or $(\varepsilon_{\mathrm{L}}\,\&\,\varepsilon_{\mathrm{u}})$ or ε_{C}. As discussed in § 3.1.1, in 3-axis machining, contour error is usually defined by ε_{L} [Ghaffari 2016, Yang 2015b, Altintas 2012, Yao 2012, Chiu 2001, Erkorkmaz 2006, Koren 1980], while $(\varepsilon_{\mathrm{L}}\,\&\,\varepsilon_{\mathrm{u}})$ are usually employed to define contour error in 5-axis machining [Li 2016, Yang 2015a, Khoshdarregi 2014, Zhang 2013, Altintas 2010]. To the best of the authors' knowledge, there is no report in which contour error is directly defined by ε_{C}. This can be understood because the tool path information $(\mathbf{C_L}, \mathbf{u})$ can be easily accessed, while the estimation of tool contact point $\mathbf{C_C}$ requires further information from the CAM system.

Another factor affecting the estimation approaches of contour error is the purpose of use in axis control strategy: whether it is used for contour error compensation technique during machining

process or for contour error pre-compensation technique in the off-line execution before the machining starts. Each control strategy has its own characteristics and requirements, thus the contour error estimation approach has to be consistent with the control strategy.

For contour error on-line compensation technique, contour error has to be estimated as fast as possible to respond to the real-time effect in on-line execution. Therefore, the estimation formula and assumption of contour error must be quite simple and/or the processor has to be powerful enough to overcome the computing burden. Within this context, some related works are proposed in the literature for both 3-axis and 5-axis machining, as follows.

In 3-axis machining, [Koren 1980] proved that contour error is proportional to the tool tracking error in machining linear paths. [Chiu 2001] proposed a task coordinate frame attached locally to the desired contour. By this way, some projections of the actual position tracking error in this moving task coordinate frame can be used to approximate contour error for feedback controller design. More accurate global task coordinate frame has been developed by [Yao 2012]. It calculates contour error exactly to its first order approximation. [Ghaffari 2016] used a Newton-based update law to estimate contour error.

In 5-axis machining, the influence of the actual tool orientation on contour error must be taken into account. [Altintas 2010] proposed two analytical models, for both tool tip contour error and tool orientation contour error, based on differential path geometry and the kinematics of the machine. [Yang 2015a] proposed that the contouring error components contributed by all axes drives are estimated through interpolated position commands and the generalized Jacobian function. [Li 2016] used the second-order Taylor series expansion and an analytical root-seeking formula to obtain an analytical method rather than an iterative one to estimate contour error and to overcome the computing burden in the real-time process.

For contour error pre-compensation, the computation burden is not a constraint anymore. Thus, the algorithm for contour error estimation has enough time to perform. However, the precision of the estimated contour error much depends on the accuracy of the dynamic model. Within this context, the authors [Yang 2015b, Khoshdarregi 2014, Zhang 2013, Altintas 2012, Prévost 2011a, Erkorkmaz 2006] estimated contour error using the availability of the programmed tool path and the simulated one, in which the later is calculated by solving the Forward Kinematic Transformation (FKT) with the simulated axis position responses obtained from the dynamic model of the machine axes. The general idea of these works is quite simple. First, the closest

segment on the programmed tool path to the simulated tool positioning is detected. Then, the contour error is formulated by ε_{L} for 3-axis machining or (ε_{L} & ε_{u}) for 5-axis machining.

In this work, the intended control strategy belongs to the class of the contour error pre-compensation technique in off-line execution. Therefore, the intended estimation approach for contour error should take the advantages of the existing approaches within this class. In particular, the calculation technique in [Erkorkmaz 2006] will be considered to formulate the intended contour error estimation approach.

According to the above survey, it can be noted that none of the existing works refer to the estimation of the effective tool contact point, in order to define the contour error by ε_{C}. This fact gives a motivation for this work, that is to propose some assumptions to estimate contour error from the tool contact point $\mathbf{C_C}$ in multi-axis machining within the context of off-line execution.

In summary, the above related studies and the intention of this work are classified by Table 1.1.

Table 1.1: Some related works for contour error estimation and ✓ the intention of this work

	3-axis machining		5-axis machining	
	ε_L	ε_C	ε_L & ε_u	ε_C
On-line	[Ghaffari 2016] [Yao 2012] [Chiu 2001] [Koren 1980]		[Li 2016] [Yang 2015a] [Altintas 2010]	
Off-line	[Yang 2015b] [Altintas 2012] [Erkorkmaz 2006]	✓	[Khoshdarregi 2014] [Zhang 2013]	✓

3.1.3 Contour error estimation based on tool contact point

According to § 3.1.1 and § 3.1.2, it can be noticed that even if both ε_{L} and ε_{u} are non-zero, the effective cutting error ε_{C} may be zero, as illustrated in Fig. 1.15.

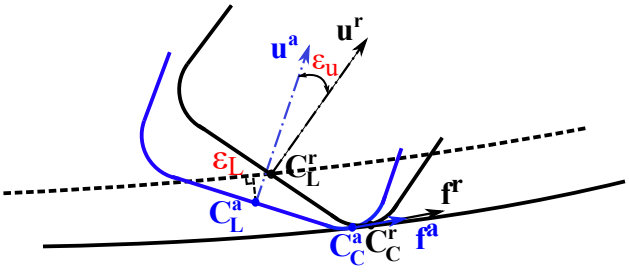

Figure 1.15: Zero cutting error in the presence of tool path deviation

The above issue gives a motivation for the following proposed contour error approximation approach. As a consequence, it is proposed to improve the contour error by controlling directly the $\mathbf{C_C}$ point and the ε_C error.

Contour error formulation. The mathematic formula of the tool contact point derives from the formulation of the tool positioning in § 2.2.

Take the case of toric tool for an example, according to (1.2) and (1.3), the tool contact point $\mathbf{C_C}$ is defined by (1.9),

$$\mathbf{C_C} = \mathbf{C_L} - r.(\mathbf{n} - \mathbf{u}) - R.\frac{\mathbf{u} \wedge \mathbf{n}}{\|\mathbf{u} \wedge \mathbf{n}\|} \wedge \mathbf{u} \tag{1.9}$$

in which the tool path information $(\mathbf{C_L}, \mathbf{u})$ can be reformulated by solving the FKT based on the position setpoints and the position responses, as found in [Lavernhe 2006]. Clearly, the tool contact point $\mathbf{C_C}$ will be determined if the unit surface normal vector $\mathbf{n}$ is known. Determination of $\mathbf{n}$ is the main difficulty to estimate the tool contact point.

As it can be seen in Fig. 1.5, as $\mathbf{u}$ is known, $\mathbf{n}$ can be reformulated if other parameters $\mathbf{f}$, θ_n and θ_t are known or estimated. To obtain the latter values, it is proposed to make some assumptions as follows:

- For both the programmed and actual tool paths, the unit feed vector $\mathbf{f}$ at $\mathbf{C_C}$, denoted as $\mathbf{f_{C_C}}$, is assumed to be equal to the unit tangential vector as a function of tool displacement at $\mathbf{C_L}$, denoted as $\mathbf{f_{C_L}}$, if the two following conditions are satisfied:
  - The tool diameter is much smaller than the trajectory curvature,
  - The tool inclination variation and its angular velocity, $\Delta\theta_t$ and $\dot{\theta}_t$ respectively, are very small, e.g. several micro-degrees.

This assumption is illustrated in Fig. 1.16.

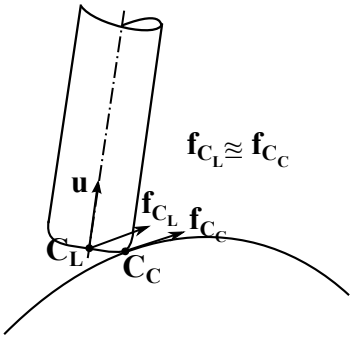

Figure 1.16: Estimation of the feed vector

When $\mathbf{C}_L$ is known, $\mathbf{f_{C_C}}$ and $\mathbf{f_{C_L}}$ are defined by the first derivative of $\mathbf{C}_L$ coordination as a function of the tool displacement s, expressed in (1.10),

$$\mathbf{f_{C_C}} \approxeq \mathbf{f_{C_L}} = \frac{d\left([\mathbf{C}_{Lx}, \mathbf{C}_{Ly}, \mathbf{C}_{Lz}]\right)}{ds} \tag{1.10}$$

where the tool displacement at the k instant, denoted as s^k, defined in (1.11).

$$s^k = \sum_{i=1}^{k} \|\mathbf{C}_{\mathbf{L}}^{r,i} - \mathbf{C}_{\mathbf{L}}^{r,i-1}\| \tag{1.11}$$

In the following, only the symbol $\mathbf{f}$ is used for the feed vector, that is defined in (1.10).

- For θ_n and θ_t, their reference values are known in CAM configuration. Assume that the responses of θ_n and θ_t can be estimated by specified values

Once all of the reference and actual values of $\mathbf{u}$, $\mathbf{f}$, θ_n and θ_t are determined, vector $\mathbf{n}$ can be calculated as follows.

According to Fig. 1.5, given the unit vectors $\mathbf{u}\,(\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$, $\mathbf{f}\,(\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z)$ and $\mathbf{n}\,(\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z)$. A unit vector $\mathbf{m}\,(\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z)$ is obtained by a rotation of $\mathbf{f}$ with an angle of θ_n around $\mathbf{n}$, having the same direction with the projection of $\mathbf{u}$ onto the plane $(\mathbf{t}, \mathbf{f})$. Its coordinates are defined by (1.12).

$$[\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z]^T = R_n\,(\theta_n)\,.\,[\mathbf{f}_x, \mathbf{f}_y, \mathbf{f}_z]^T \tag{1.12}$$

where $R_n(\theta_n)$, a rotation matrix around $\mathbf{n}$ [Kovács 2012], is given in (1.13),

$$R_n(\theta_n) = \begin{bmatrix} c + \mathbf{n}_x^2(1-c) & \mathbf{n}_x\mathbf{n}_y(1-c) - \mathbf{n}_z s & \mathbf{n}_x\mathbf{n}_z(1-c) + \mathbf{n}_y s \\ \mathbf{n}_y\mathbf{n}_x(1-c) + \mathbf{n}_z s & c + \mathbf{n}_y^2(1-c) & \mathbf{n}_y\mathbf{n}_z(1-c) - \mathbf{n}_x s \\ \mathbf{n}_z\mathbf{n}_x(1-c) - \mathbf{n}_y s & \mathbf{n}_z\mathbf{n}_y(1-c) + \mathbf{n}_x s & c + \mathbf{n}_z^2(1-c) \end{bmatrix} \tag{1.13}$$

with $c = cos(\theta_n)$ and $s = sin(\theta_n)$.

Due to the fact that $\mathbf{n}$ is perpendicular to $\mathbf{f}$, and the definition of θ_t and θ_n, the relations in (1.14) are obtained.

$$\begin{cases} \mathbf{n} \perp \mathbf{f} \\ \angle(\mathbf{n}, \mathbf{u}) = \theta_t \\ \angle(\mathbf{m}, \mathbf{u}) = 90 - \theta_t \end{cases} \tag{1.14}$$

From (1.14), leading to (1.15).

$$\begin{cases} \mathbf{n}.\mathbf{f} = 0 \\ \mathbf{n}.\mathbf{u} = \|\mathbf{n}\|.\|\mathbf{u}\|.cos(\theta_t) \\ \mathbf{m}.\mathbf{u} = \|\mathbf{m}\|.\|\mathbf{u}\|.cos(90 - \theta_t) \end{cases} \tag{1.15}$$

As $\mathbf{u}$, $\mathbf{f}$, $\mathbf{n}$ and $\mathbf{m}$ are unit vectors, from (1.15), (1.16) is deduced.

$$\begin{cases} \mathbf{n}_x.\mathbf{f}_x + \mathbf{n}_y.\mathbf{f}_y + \mathbf{n}_z.\mathbf{f}_z = 0 \\ \mathbf{n}_x.\mathbf{u}_x + \mathbf{n}_y.\mathbf{u}_y + \mathbf{n}_z.\mathbf{u}_z = cos(\theta_t) \\ \mathbf{m}_x.\mathbf{u}_x + \mathbf{m}_y.\mathbf{u}_y + \mathbf{m}_z.\mathbf{u}_z = sin(\theta_t) \end{cases} \tag{1.16}$$

By solving (1.16), the coordinate components of $\mathbf{n}$ can be obtained.

To verify the above assumption, a test case is proposed with the yaw angle θ_n sets to zero and the tilt angle θ_t sets to 5 degrees. The previous equation (1.16) can be simplified by (1.17).

$$\mathbf{n} = \frac{\mathbf{f} \wedge \mathbf{u}}{\|\mathbf{f} \wedge \mathbf{u}\|} \wedge \mathbf{f} \tag{1.17}$$

Once $\mathbf{n}$ is estimated by (1.17), $\mathbf{C_C}$ is estimated by (1.9).

Calculation technique. To fulfill the proposed contour error (ε_C) estimation, the orthogonal distance from the actual tool contact point $\mathbf{C_C^a}$ to the reference $\mathbf{C_C}$ path must be found, as illustrated in Fig. 1.13.b.

The calculation technique of the contour error in off-line execution is based on [Erkorkmaz 2006], whose main ideas are illustrated in Fig. 1.17 and summarized below.

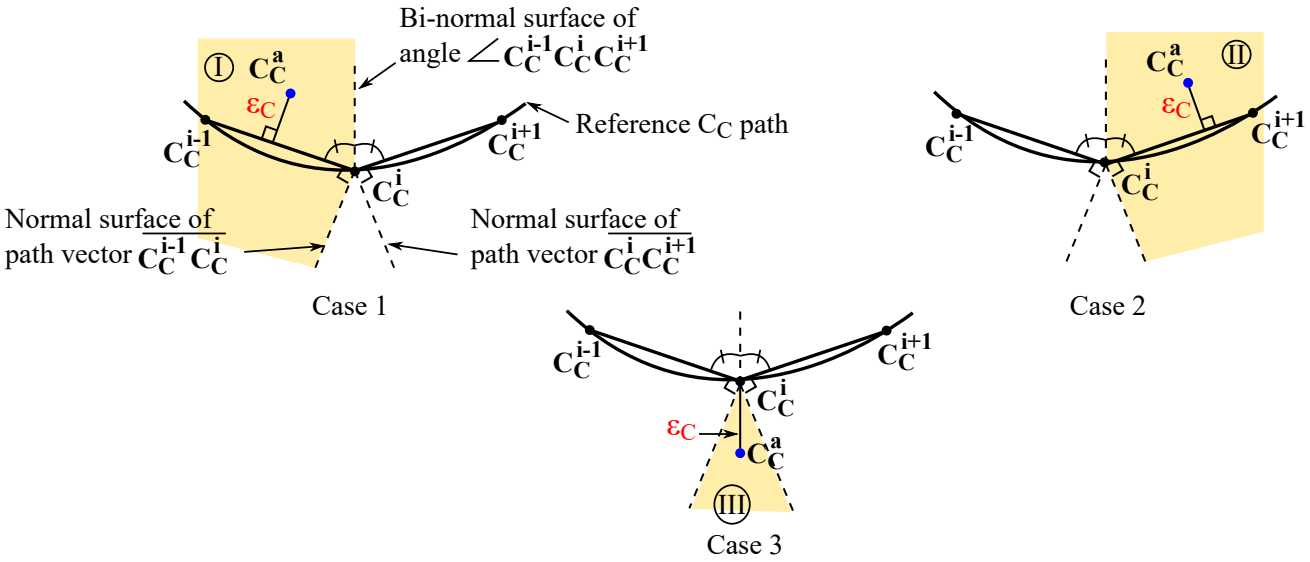

Figure 1.17: Three cases for contour error estimation

- The actual tool contact point $\mathbf{C_C^a}$ can belong to one of the three regions, I, II and III, which are bounded by the normal surfaces of the path segment vectors and the bi-normal surface of the angle $\angle \mathbf{C_C^{i-1} C_C^i C_C^{i+1}}$.

- Case 1: Contour error, ε_C, is estimated using the previous closest $\mathbf{C_C}$ segment, $\overline{\mathbf{C_C^{i-1} C_C^i}}$, and given in (1.18).

$$\varepsilon_C = distance_{\perp}\left(\mathbf{C_C^a}, \overline{\mathbf{C_C^{i-1} C_C^i}}\right) = \frac{\|\overrightarrow{\mathbf{C_C^{i-1} C_C^i}} \times \overrightarrow{\mathbf{C_C^{i-1} C_C^a}}\|}{\|\overrightarrow{\mathbf{C_C^{i-1} C_C^i}}\|} \tag{1.18}$$

- Case 2: Contour error, ε_C, is estimated using the next closest $\mathbf{C_C}$ segment, $\overline{\mathbf{C_C^i C_C^{i+1}}}$, and given in (1.19).

$$\varepsilon_C = distance_{\perp}\left(\mathbf{C_C^a}, \overline{\mathbf{C_C^i C_C^{i+1}}}\right) = \frac{\|\overrightarrow{\mathbf{C_C^i C_C^{i+1}}} \times \overrightarrow{\mathbf{C_C^i C_C^a}}\|}{\|\overrightarrow{\mathbf{C_C^i C_C^{i+1}}}\|} \tag{1.19}$$

- Case 3: Contour error, ε_C, is estimated using the closest reference tool contact point $\mathbf{C}_\mathbf{C}^\mathbf{i}$, and given in (1.20).

$$\varepsilon_C = \|\overrightarrow{\mathbf{C}_\mathbf{C}^\mathbf{a}\mathbf{C}_\mathbf{C}^\mathbf{i}}\| \tag{1.20}$$

From (1.18) to (1.20), the absolute value of contour error, ε_C, is obtained. To evaluate whether contour error represents an under cut or an over cut, the sign of contour error is a good indicator [Prévost 2011a]. It is proposed that the positive and negative signs of contour error represent for an under cut and an over cut, illustrated in Fig. 1.18.a and b, respectively.

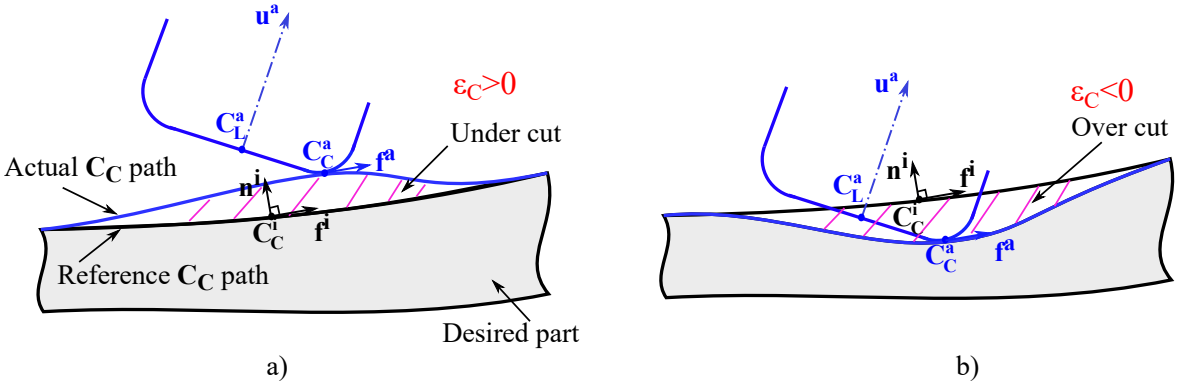

Figure 1.18: Illustration of contour error sign

The sign of contour error is mathematically defined in (1.21).

$$sign\left(\varepsilon_C\right) = sign\left(\mathbf{n}^\mathbf{i}.\overrightarrow{\mathbf{C}_\mathbf{C}^\mathbf{i}\mathbf{C}_\mathbf{C}^\mathbf{a}}\right) \tag{1.21}$$

Validation. It is clear that the accuracy of the estimated contour error, from (1.18) to (1.20), belongs to the accuracy of the estimated tool contact point in (1.9). The latter belongs to the accuracy of $\mathbf{n}$ in (1.17).

To validate $\mathbf{n}$, it is proposed to perform the two following steps:

- Firstly, from the estimated $\mathbf{f}$ and estimated $\mathbf{n}$, calculate the estimated tilt angle θ_t,
- Then, compare the estimated θ_t with the programmed one.

Note that in the case of study, the programmed tool path generated by CAM has the specificity of tilt and yaw angles as $\theta_t = 5\ deg$ and $\theta_n = 0\ deg$, respectively.

Fig. 1.19 shows the programmed tool path, the estimated $\mathbf{C_C}$ path and the estimated vectors.

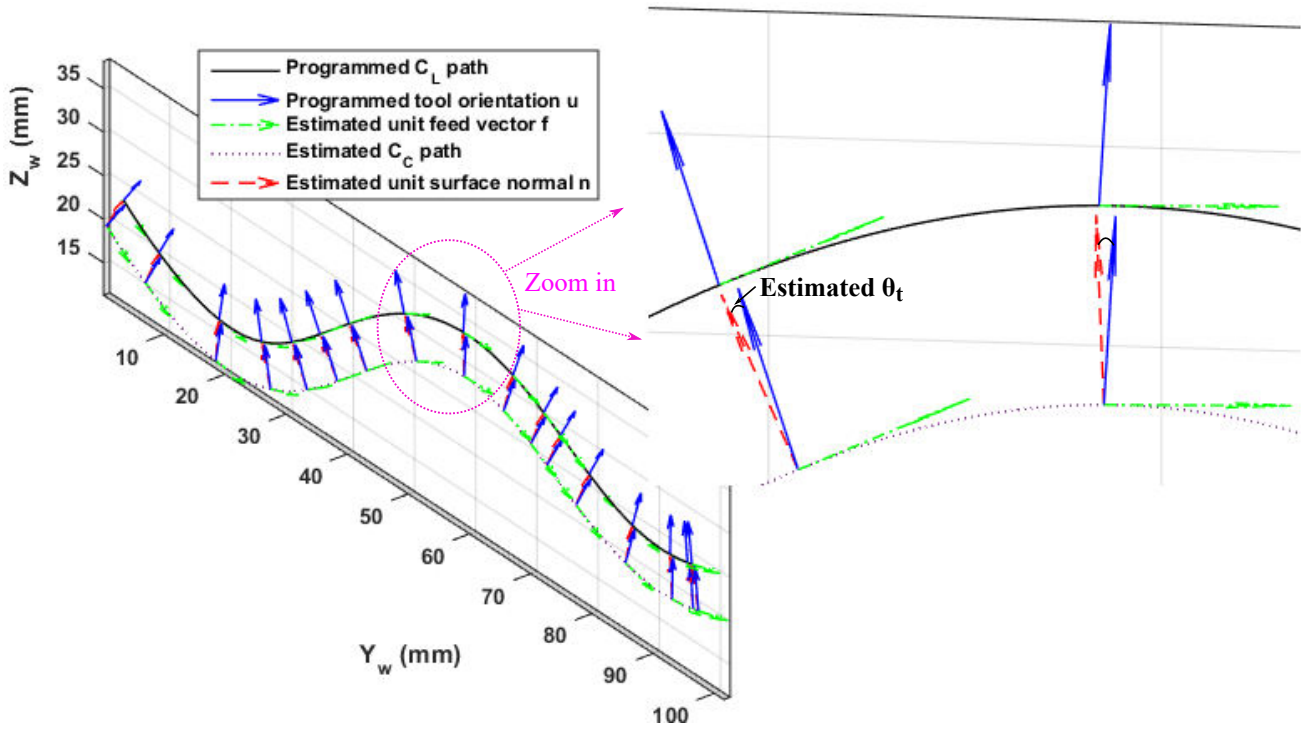

Figure 1.19: Illustrations of the programmed tool path, the estimated $\mathbf{C_C}$ path and the estimated vectors

As it can be seen in Fig. 1.20, the estimated θ_t values accurately approximates the programmed value, with a tolerance of less than 1.10^{-4} degrees. Therefore, the approximation of vector $\mathbf{n}$ has a high confidence, leading to the high precision of the estimated $\mathbf{C_C}$.

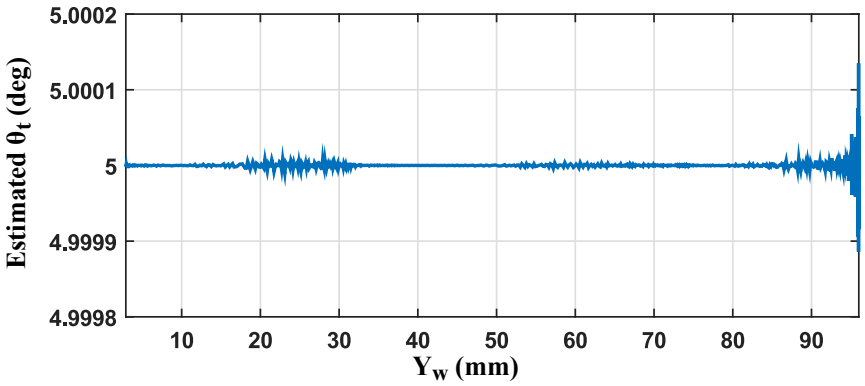

Figure 1.20: Verification of θ_t value

3.2 Contour error estimation in 5-axis flank milling

The general 5-axis flank milling is illustrated in Fig. 1.21. As it can be seen, there exist the over cut and/or under cut due to the deviations of the higher and lower parts of the tool trunk over the workpiece.

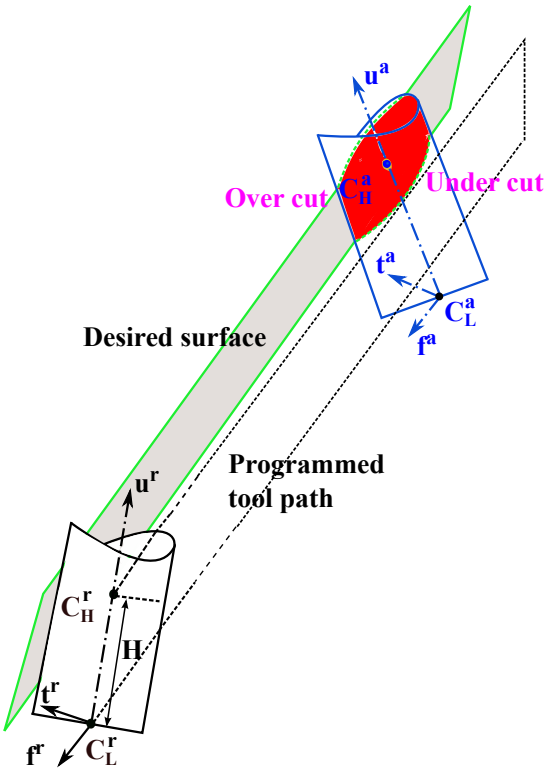

Figure 1.21: Over cut and under cut in flank milling

Controlling the higher and lower parts of the tool trunk is performed through either controlling the programmed tool path $(\mathbf{C_L}, \mathbf{u})$, or controlling two points: one for the programmed tool location point $\mathbf{C_L}$ and another point on the tool trunk, denoted as the higher tool location point $\mathbf{C_H}$. This means that controlling a couple of values $(\mathbf{C_L}, \mathbf{C_H})$ is equivalent to controlling the programmed tool path $(\mathbf{C_L}, \mathbf{u})$. These two points are characterized by (1.22),

$$\mathbf{C_L C_H} = \mathbf{H u} \tag{1.22}$$

where $\mathbf{H}$ is the distance between $\mathbf{C_L}$ and $\mathbf{C_H}$.

3.2.1 Problem formulation

In 3-axis flank milling, due to the fixed tool orientation, only a limited number of surfaces can be machined, e.g. in flank milling with a 3-axis machine having a vertical tool axis, the walls with fixed orientations should be parallel to the machine tool axis. Fig. 1.22 illustrates that the errors in controlling the tool location point generate an under cut and an over cut over the workpiece.

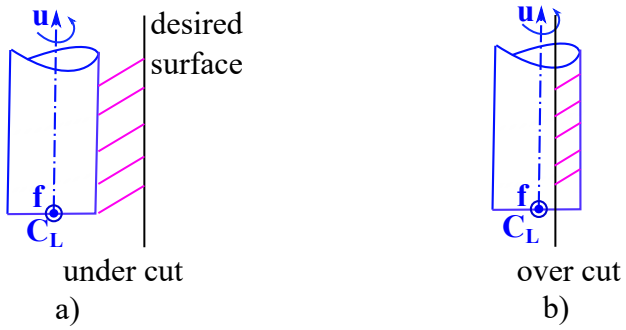

Figure 1.22: Impact of tool location error on the effective cutting errors of 3-axis flank milling

In 5-axis flank milling, the effect of the tool location point error on the effective cutting error is the same as in the 3-axis flank milling case. However, in this case, the errors of the tool orientation vector also have a high impact on the effective cutting error, as illustrated in Fig. 1.23.

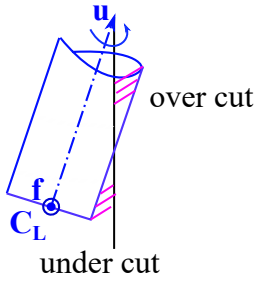

Figure 1.23: Impact of tool orientation error on the effective cutting error of 5-axis flank milling

The definition of contour error in flank milling consists in approximating the above effective cutting errors between the nominal surface and the machined surface.

The literature review related to this topic is briefly presented below.

3.2.2 Literature review

Contour error in flank milling is usually determined by calculating the surface envelope of the tool movement, as illustrated in Fig. 1.7.

To determine the surface envelope, each author proposed their own approach, that generally belongs to one of the two following groups. The first one is the analytical approach, that refers to calculating the implicit equation of the envelope, such as using "Jacobian Rank Deficiency"[Abdel-Malek 1997, Yang 2005], or "Sweep Differential Equation"[Blackmore 1994], etc. The second one is the kinematic approach, that consists in estimating the surface envelope, such as using "Tangency fonction"[Chiou 2004], "Grazing Curve"[Bedi 2003], etc. While the analytical approach is generally complex, time-consuming and difficult to be implemented, the kinematic approach is quite simple to obtain quickly the approximated surface envelope. That is the reason why the latter will be preferred in this thesis.

Because proposing a new estimation approach for the contour error of flank milling is not the objective of this thesis, the calculation technique of this error in the next paragraph is based on a kinematic approach proposed in the literature, that is [Pechard 2011].

3.2.3 Calculation technique

Similar to the calculation technique of contour error in point milling in § 3.1.3, it is possible to find out the reference $\mathbf{C_L}$ path segment $\overline{\mathbf{C_L^i C_L^{i+1}}}$, that is closest to the actual tool positioning $(\mathbf{C_L^a}, \mathbf{C_H^a})$, illustrated in Fig. 1.24.

From $\mathbf{C_L^i}$, one can define a specified local plane $\mathbf{P^i}$, defined by (1.23),

$$\mathbf{P^t} = \left(\mathbf{C_L^i}, \mathbf{u^i}, \mathbf{f^i}\right) \tag{1.23}$$

where the unit feed vector $\mathbf{f^i}$ is similarly defined in (1.10).

The estimated contour errors generated by the lower and higher tool location points, $\mathbf{C_L}$ and $\mathbf{C_H}$, denoted by ε_L and ε_H, are provided by (1.24) and (1.25), respectively,

$$\varepsilon_L = distance_{\perp}\left(\mathbf{C_L^a}, \mathbf{P^i}\right) = \|\overrightarrow{\mathbf{C_L^a N}}\| \tag{1.24}$$

$$\varepsilon_H = distance_{\perp}\left(\mathbf{C_H^a}, \mathbf{P^i}\right) = \|\overrightarrow{\mathbf{C_H^a M}}\| \tag{1.25}$$

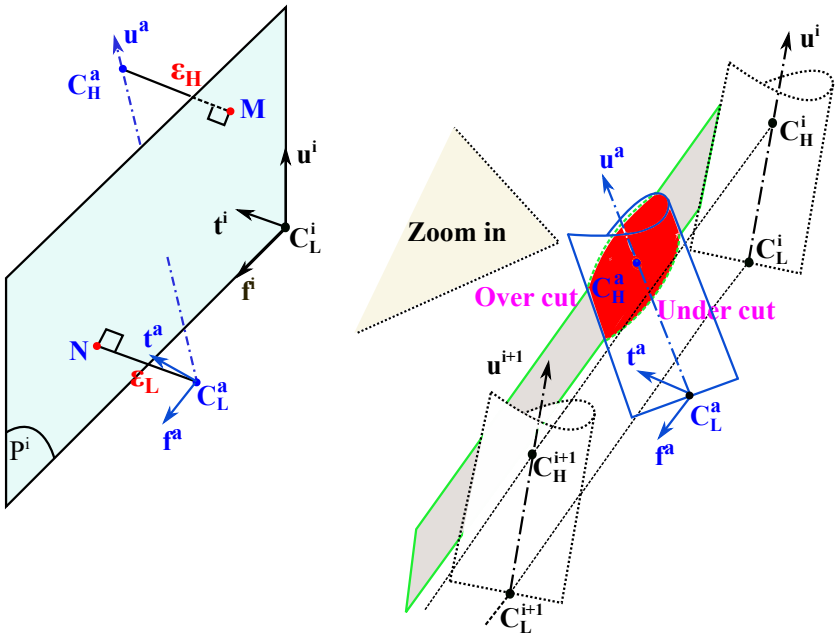

Figure 1.24: Contour error definition in flank milling

where $\mathbf{N}$ and $\mathbf{M}$ are the projections of $\mathbf{C_L^a}$ and $\mathbf{C_H^a}$ on the plane $\mathbf{P^i}$, respectively. According to (1.24) and (1.25), contour error of 5-axis flank milling, denoted as ε_F, is finally estimated by (1.26).

$$\varepsilon_F = max\left(\varepsilon_L, \varepsilon_H\right) \tag{1.26}$$

3.3 Conclusion

To conclude, the estimated contour errors for both point milling and flank milling in multi-axis machining center have been formulated in this section.

For point milling, the primary idea of the proposed contour error estimation approach is based on the estimation of the tool contact point.

Meanwhile, in flank milling the estimation of the contour error is still based on the evaluation of the tool location points.

4 Feedrate interpolation

Another execution before controlling the machine axes is feedrate interpolation. This stage is important, because it generates the position setpoints, that are the input of the axis control stage.

It can be noted that the position response of each axis, which is the output of axis control stage, is expected to follow the position setpoint as close as possible. Thus, the behaviors of the position setpoint, i.e. its kinematic characteristics, will affect the behaviors of the position response as well. Consequently, the latter affects the tool and machining behaviors. For example, if the reference axis jerk exceeds the jerk limit of machine axis, its response may have a high possibility to exceed this limitation also. This will induce the tool vibration during machining, causing marks on the machined part.

Henceforth, in developing an advanced axis control law for the contour error reduction, it is important to take the constraints of the position setpoints into account, in the way that the resulting position responses should also respect such constraints.

To evaluate the constraints of the position setpoints, it is necessary to intrinsically understand the core idea of the feedrate interpolation.

4.1 Overview

The schematic diagram of feedrate interpolation is highlighted in Fig. 1.25, which is explained below.

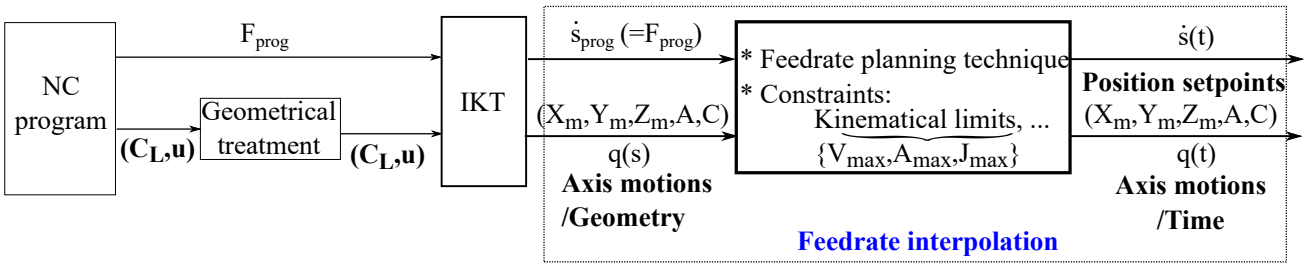

Figure 1.25: Schematic diagram of feedrate interpolation

Firstly, the NC program contains the programmed tool path $(\mathbf{C_L}, \mathbf{u})$ and the programmed fee-

drate F_{prog}. While the former is geometrically treated by the CNC, such as rounding off the discontinuities in the programmed tool path described by G1 code, the latter is always constant, determined by the cutting conditions between the tool and workpiece material. They are the two inputs for the IKT, to determine the joint movements.

Next, the programmed feedrate and the joints movements are evaluated as a function of geometry, namely denoted $\dot{s}_{prog}$ and $q(s)$, where s represents the tool displacement along the tool path. The two data constitute the inputs of the feedrate interpolation. The outputs of this stage are the feedrate and the sampled axis position setpoints, evaluated as a function of time, and denoted $\dot{s}(t)$ and $q(t)$, respectively. The core part of the feedrate interpolation is primarily based on the feedrate planning technique with the respect of the concerned constraints, e.g. the axis kinematic limitations.

It can be said that CAM produces the programmed tool path typically without regards to the constraints of the machine, such as axis kinematic limits, machine vibration, etc. Therefore, the purpose of the feedrate interpolation is to generate a feedrate profile to follow the given trajectory, while complying with all of these constraints. Although the programmed feedrate is always constant, the one produced by the feedrate interpolation stage is often decreased to respect all of the concerned constraints.

Some studies related to this topic are briefly reviewed as follows.

4.2 Literature review

To perform the feedrate interpolation stage, there are many proposed techniques in the literature. Each one solves its main problems and obtains its own objectives.

[Bobrow 1985] solved the minimum time control problem based on the solution from the feedrate interpolation stage, in which the trajectory generation is considered as a dynamic system with two states, including path displacement and velocity; then the velocity along the tool path is limited by the actuator torque constraints. Moreover, the tool path acceleration is switched between its maximum and minimum limits at the identified path points to generate a bang-bang style trajectory.

Furthermore, there are problems associated to the feedrate planning along a curved path, be-

ing subject to bounds on the acceleration to achieve minimum traversal time in machining [Shin 1985, Shiller 1990]. [Timar 2005] has addressed such problems for a 3-axis CNC machine, in which by using polynomial parametric curves, the optimal feedrate can be specified by a piecewise-rational function of the parameter. Consequently, a simple real-time interpolator algorithm, directly driving the machine based on the analytic path description, can be performed to obtain the optimized feedrate, without the need for piecewise-linear/circular G code approximations.

To remove feedrate fluctuations due to parametrization errors and to produce continuous profiles of position, velocity, and acceleration, [Erkorkmaz 2001a] proposed a quintic spline interpolation approach, in which fifth order polynomials are used to re-sample the reference trajectory generated with varying interpolation period at the servo loop closure period. This allows preservation of the original kinematic profiles. In addition, to counteract with the feedrate planning for long tool paths, a Linear Programming combined with a parallel windowing algorithm is proposed in [Erkorkmaz 2017], from which the interpolation can be performed over different portions along the tool paths in parallel, with minimal impact on the optimality of the obtained feedrate profile.

On top of this, it can be said that the axis kinematic constraints, including the velocity, acceleration and jerk limitations of the machine axes, are usually considered in the feedrate interpolation stage. This is due to the fact that such constraints allow preservation of the operational life of the drives and of course of the machine, avoidance of saturations, and the prevention of machine vibrations that can potentially lead to severe marks on the surface or instability of the axis control [Sencer 2008, Beudaert 2012]. The velocity and acceleration limitations of the drives in the feedrate interpolation are carried out in [Dong 2005, Renton 2000]. Indeed, jerk is also an important parameter, because it allows to limit the feedrate variations in high speed machining. [Erkorkmaz 2001a, Barre 2005] have highlighted the effect of jerk limitation on the mechanical structure.

4.3 Chosen algorithm

Developing a new technique of feedrate interpolation is out of the scope of this work. Therefore, it is proposed to use one of the proposed feedrate interpolation technique in the literature to generate the position setpoint for the axis control. In fact, the chosen technique for the feedrate interpolation stage is the VPOp algorithm proposed in [Beudaert 2012], which has been

developed at the laboratory LURPA. Main characteristics of VPOp are highlighted as follows:

- To reduce the frequency content of the trajectory and to avoid exciting the natural modes of the structure, the limitation of not only the tangential jerk but also of each axis jerk should be taken into consideration.

- The geometrical problem is separated from the temporal interpolation problem by a decoupled approach. The interpolation in sharp zones is based on a look ahead algorithm, in considering the geometrical analysis of the tool path.

- Its mathematical formulation allows processing both linear and rotary axes in the same manner and the algorithm is suitable for either serial or parallel kinematic machines.

The main equation representing the principle of VPOp algorithm, given in (1.27), expresses the upper limits of the feedrate $\dot{s}$ subject to the minimum value between the programmed feedrate and the upper feedrate limits derived from the limitations of axis velocity, acceleration and jerk.

$$\dot{s} \leq \min \left(F_{prog}, \left(\frac{V^{max}}{|\mathrm{q}_s|} \right), \left(\sqrt{\frac{A^{max}}{|\mathrm{q}_{ss}|}} \right), \left(\sqrt[3]{\frac{J^{max}}{|\mathrm{q}_{sss}|}} \right) \right) \tag{1.27}$$

The equation (1.27) allows to predict the areas where the real feedrate will decrease, while respecting all of the kinematic constraints.

The resulting feedrate obtained by VPOp is illustrated in Fig. 1.26. As it can be seen, the feedrate respects all the constraints in (1.27).

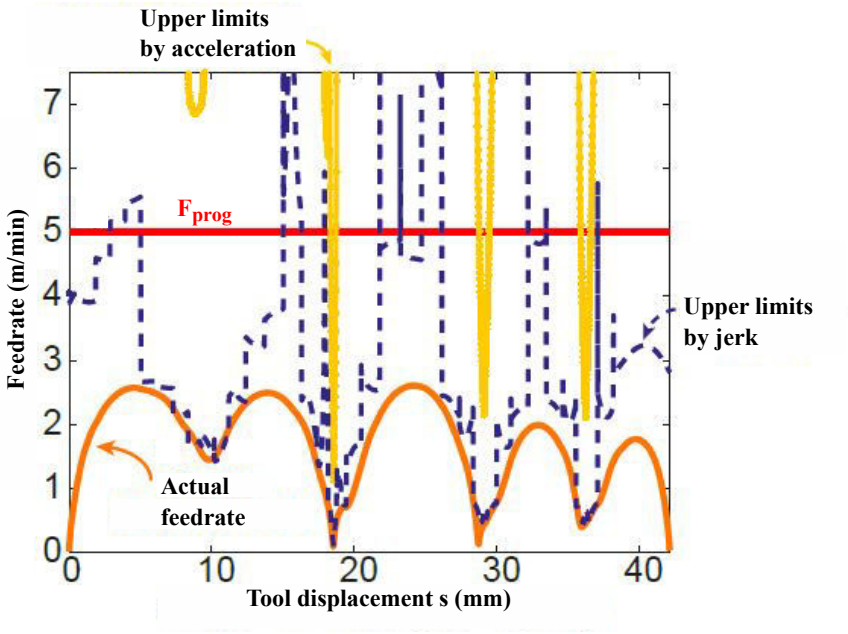

Figure 1.26: Resulting feedrate by VPOp [Beudaert 2013]

5 Conclusions

The tool path geometry and the kinematic execution in multi-axis high speed machining have been analyzed in this chapter.

Concerning the geometry treatments, firstly the necessary knowledge in computing the multi-axis tool path have been given. Moreover, the issues of the real tool path derived from the changes of the tool orientation in 5-axis milling contexts have been discussed. The contour error estimation approaches in 5-axis point milling and 5-axis flank milling have been formulated. The former is proposed, based on the estimation of the tool contact point through some assumptions. The latter is based on one of the proposed approach in the literature.

Relating to the kinematic execution, the important roles of feedrate interpolation for the axis control and for the machining are highlighted. Furthermore, the principle and purpose of this task are discussed. The chosen feedrate interpolation technique generates the position setpoint respecting the axis kinematic limitations of the machine. Therefore, the adaptive control law in the next chapter should be developed in the way that the position responses respect these constraints as well.

Finally, it can be concluded that this chapter has successfully provided the necessary elements (the position setpoint generation, the contour error formulation, the axis kinematic constraints) so that the intended adaptive control law for the contour error reduction can be developed.

Chapter 2

Adaptive contouring control in multi-axis high speed machining

The previous chapter has formulated the contour error estimation approach within the different contexts of multi-axis high speed machining. By using the nonlinear axis model, the contour error can be predicted by means of an off-line process.

This chapter aims at solving an optimization problem to produce the optimal variable control gains in pre-compensating the contour error, before it occurs in the machining process.

The introduction of the classical axis control structure is firstly presented. The nonlinear feature of the axis model is then highlighted. Finally, the core part of this chapter is to formulate and solve the optimization problem under constraints.

1 Introduction of the classical axis control and the nonlinear axis model

The axis control in CNC is built under the classical cascaded structure [Altintas 2000a]. It consists of the position, velocity and current loops, ranging from the outer to inner loops, respectively, illustrated in Fig. 2.1.

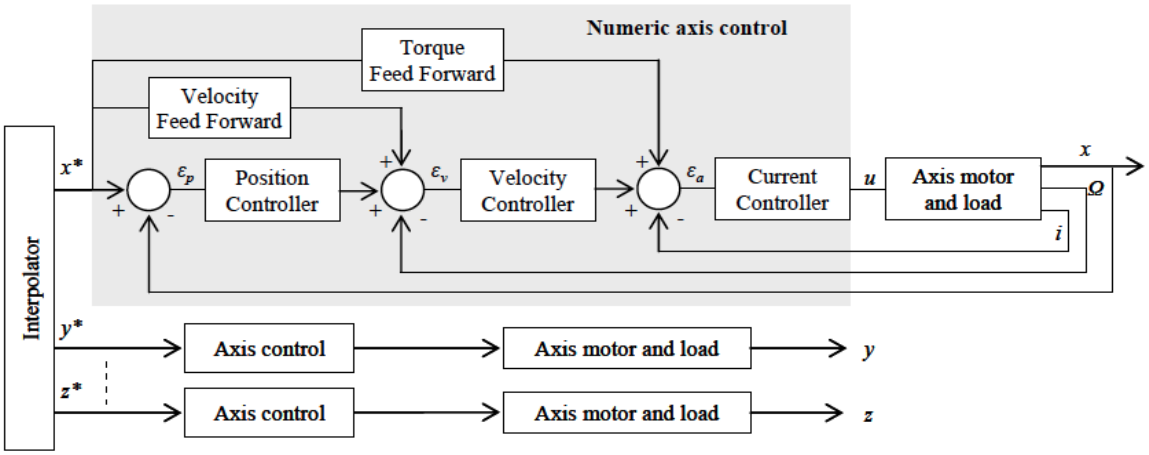

Figure 2.1: Axis control structure [Susanu 2005]

The sampling time decreases from the position to current loops. The position loop is equipped with a Proportional (P) controller and a velocity Feed Forward (FFW) action. A Proportional-Integral (PI) controller with a torque FFW action is used in the velocity loop, while the current

loop uses only a PI controller. The purpose of these controllers is to reduce the axis tracking error, with an expectation that the contour error is also reduced during the machining process.

In this thesis, due to the aim of contour error pre-compensation in an off-line phase, the nonlinear axis model of the above classical axis control is used to investigate the proposed control law. For the case of study, it is proposed to use the nonlinear axis model of the Mikron UCP 710 machine tool located at LURPA, that has been developed in [Susanu 2005], and enhanced and validated in [Prévost 2011a]. Obviously, the proposed control approach can be used for other axis models of other machining centers. It can be found in [Prévost 2011a], the chosen model has been proved to simulate the real behavior in 5-axis milling within Mikron UCP 710 machining center with high confidence. One validation example is for machining the impeller surface illustrated in Fig. 2.2, whose the simulated and measured axis tracking errors are superimposed almost at all portions of the machining profiles, as seen in Fig. 2.3.

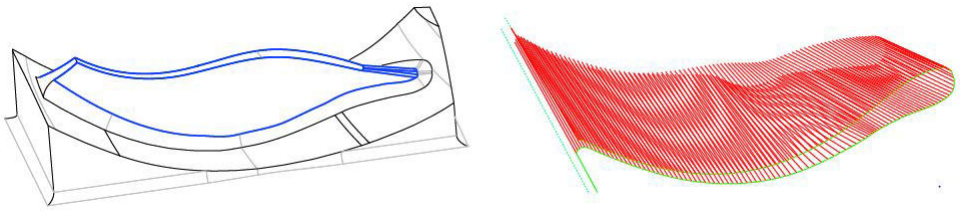

Figure 2.2: Definition of impeller surface and the machining trajectory [Pechard 2011]

In the selected axis model, only the nonlinear friction characteristic is taken into account. The friction law in each axis drive is studied and proposed in [Prévost 2011a] and illustrated in Fig. 2.4. It is a combination of both viscous and Coulomb frictions.

The detailed descriptions of the friction model as well as the motor drive model are given in Appendix A.

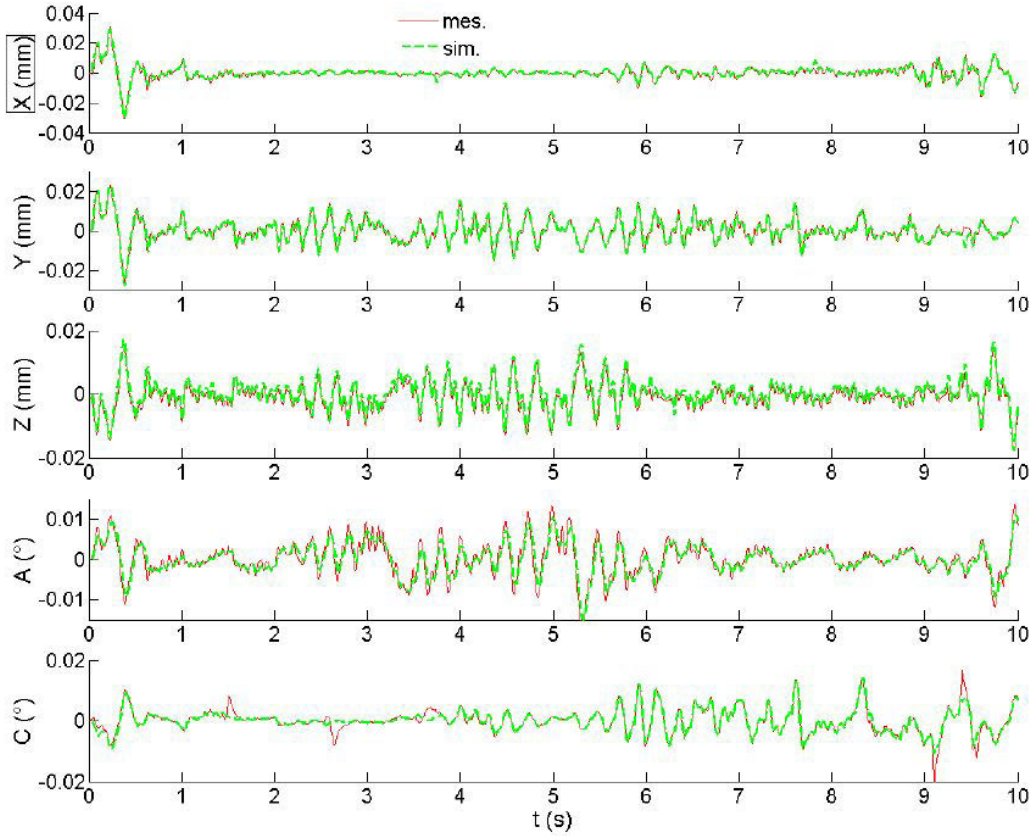

Figure 2.3: Simulated and measured axis tracking errors of the impeller surface [Prévost 2011a]

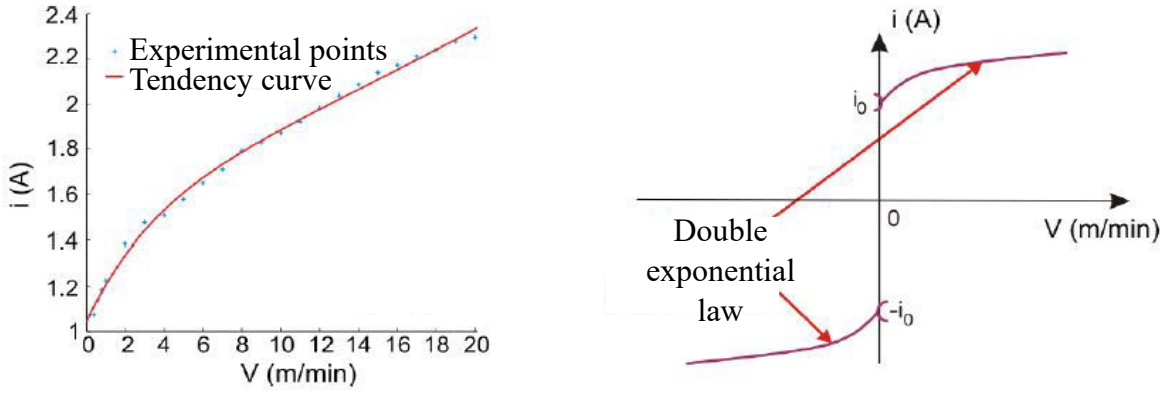

Figure 2.4: Friction law and experimental results for X axis [Prévost 2011a]

2 Contouring control: State of the art

The fact is that the presence of the contour error definitely indicates the existence of the tool tracking error, the opposite is not always the case. This means that although the tool tracking error may exist, the contour error can be zero. It is well-known that obtaining small contour error is more important than small tool tracking error. This is because the contour error is usually referred to the conformity between the machined and desired parts. The control efforts for contour error reduction can be called contouring control [Erkorkmaz 2001b]. There are two main techniques in order to deal with such a control, including contour error compensation and pre-compensation techniques.

For the contour error compensation, advanced axis controllers are usually developed. There are two sub-philosophies in this way.

On the one hand, authors proposed advanced controllers for axis tracking. The motivation is that when the axis tracking error is reduced in all of the axes, it may indirectly lead to the contour error reduction.

Altintas et al. [Altintas 2000b] built a sliding mode controller to reduce the tracking error of each axis in the presence of external disturbances such as friction and cutting force, obtaining an indirect reduction of contour error. Erkorkmaz et al. [Erkorkmaz 2001b] used a pole placement controller with disturbance cancellation in the feedback loop, in order to deal with the detrimental effects of friction, cutting forces, and drive parameter variations. A combination of vibration avoidance, sliding mode control, torque ripple and friction compensation techniques to facilitate high bandwidth – high accuracy tracking in ball screw drives is proposed in [Kamalzadeh 2007]. The model predictive control is also proposed for the axis control in the context of high speed machining in [Rodriguez-Ayerbe 2014, Dumur 2008].

The benefit of this method is that the idea is simple, as it is based on the axis tracking error reduction to reduce the contour error. Moreover, the tracking error is easily obtained during the control process thanks to the measuring devices. However, it should be noted that the decrease of tracking error, which is still larger than zero, may be less effective in reducing the contour error or can even increase the contour error in some cases of multi-axis machining [Koren 1991, Tang 2013].

On the other hand, to directly reduce the contour error, especially in the case of multi-axis

machining, authors proposed the advanced contouring controllers. In this way, all axis tracking error and position responses are collected and used to predict the contour error by one of the estimation methods. The control signal of each axis is then modified by one of the intelligent control laws to compensate the contour error [Koren 1991, Chiu 2001].

Following this philosophy, Koren [Koren 1980, Koren 1991] was a pioneer of Cross Couple Control (CCC) for compensating the contour error online. Afterwards, a large number of extended and/or improved versions of CCC are proposed [Huo 2012, Zhang 2015]. Cheng et al. [Cheng 2009] proposed an integrated motion control scheme combined with a fuzzy logic-based feedrate regulator, which adjust the value of the desired feedrate, to reduce the contour error. In addition, Khalick et al. [El-Khalick 2011] added the real time estimated contour error into the optimization problem of Model Predictive Control (MPC), in yielding the contour error reduction. Adaptive robust control is also proposed for biaxial contouring control in [Yao 2012, Davis 2015].

The benefit of the advanced contouring control approaches is that they can evaluate and control the real behaviors of the machine. The axis drive dynamics can be excited intelligently to compensate the contour error. In contrast, the drawback is the computing load in real time, especially in 5-axis machining. Moreover, its effectiveness much depends on the precision of the contour error estimation method. Even if the contour error can be estimated accurately, the advanced contouring controllers require modification of the classical control structure of the commercial CNC. That may be inconvenient and costly for machine tool manufacturers.

For the contour error pre-compensation techniques, the contour error is firstly predicted based on the dynamic axis model and one of the contour error estimation approaches. Then calibration efforts are performed to pre-compensate the predicted contour error.

To the best of the authors' knowledge, almost all of the works in the literature are dedicated to modifying the position setpoints in pre-compensating the contour error on the machining operation. Khoshdarregi et al. [Khoshdarregi 2014] proposed generating the shaping position commands in the FFW blocks that pre-compensate the resulting contour error and avoid structural vibrations in 5-axis CNC machine tools. Zhang et al. [Zhang 2013] modified the trajectory commands in pre-compensating the contour error and respecting axis kinematic constraints, which are axis velocity, acceleration and jerk limitations. Yang et al. [Yang 2015b] used the idea of MPC to adjust the position setpoints, preserving the axis velocity and acceleration constraints. Furthermore, the contouring accuracy in corner machining application is also improved

by a trajectory planning strategy developed in [Erkorkmaz 2006].

The advantage of the contour error pre-compensation techniques is that the contour error can be calculated off-line through the nonlinear axis model and the computational load can be less critical. The modified position is easily implemented by users in CNC. However, this method requires strictly an accurate knowledge of the axis dynamic model obtained from the identification task.

According to the above discussions, it is proposed in this work to investigate a new strategy within the class of the contour error pre-compensation techniques. The proposed idea is that the control gains are adjusted in the off-line process through the nonlinear axis model, with an objective of the contour error reduction during the machining process. This proposed approach, so-called "Off-line Gain Adjustment" (OGA), is developed in detail below.

The works related to this topic in the literature and the intention of this thesis can be categorized by their different strategies through Table 2.1.

Table 2.1: Contouring control methods and ✓ the intention of this work

	Controller		Trajectory
	Tracking following	**Contouring following**	$(C_L, \mathbf{u})$
On-line	Sliding mode control [Kamalzadeh 2007], [Altintas 2000b], Zero Phase Error Tracking Controller [Erkorkmaz 2001b], etc.	Cross couple control [Koren 1980, Koren 1991] [Huo 2012, Zhang 2015], Fuzzy logic control [Cheng 2009], Model predictive control [El-Khalick 2011], Adaptive robust control [Yao 2012],[Davis 2015], etc.	
Off-line	Model predictive control [Rodriguez-Ayerbe 2014] [Dumur 2008], etc.	✓ **Off-line gain adjustment (OGA)**	MPC's idea[Yang 2015b], FFW shaping[Khoshdarregi 2014], Analytical method and kinematic constraints [Zhang 2013], Trajectory planning strategy [Erkorkmaz 2006], etc.

3 Off-line gain adjustment (OGA)

According to the pros and cons of the above two main contouring control techniques, it is proposed to take the advantages of both methods to compensate their drawbacks. The idea is to maintain the classical cascaded control structure and the position setpoints in the current commercial CNC, while using variable gains for the position controller, e.g. to adaptively excite the axis drive dynamic, to compensate the contour error during the machining process. The only modification on the commercial CNC is the extended functions, that are the gain update at each sampling time and the required memory to store the gain values for the given trajectory, as illustrated in Fig. 2.5.

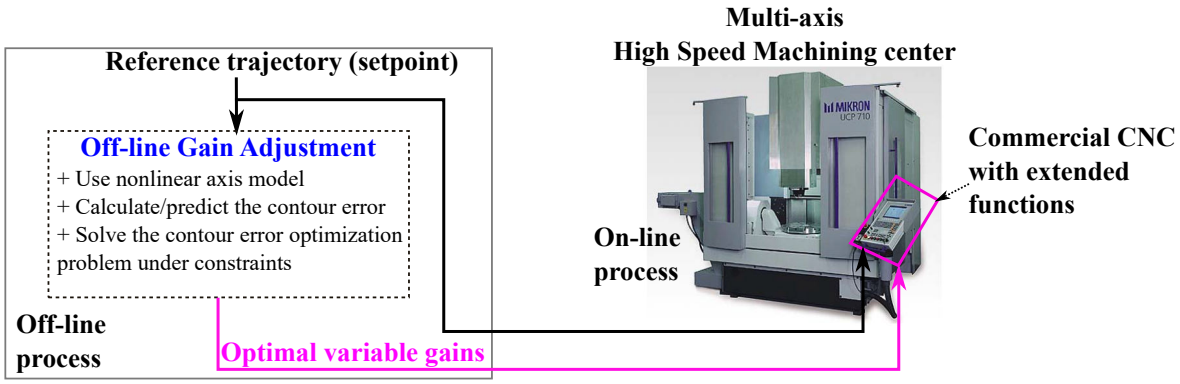

Figure 2.5: Proposed offline execution

3.1 Gain modification influence in the contour error

In the proceeding, the cases of study are related to the 5-axis point milling on the 5-axis Mikron UCP 710 machine, having three linear axes $\{X, Y, Z\}$ and two rotary axes $\{A, C\}$. For the simplicity of mathematic formulation, the following symbols are used:

- a: the axes $\{X, Y, Z, A, C\}$,
- r and s: the reference and simulated values in view of the axis model simulation, respectively.

The core part of the proposed axis control is the OGA algorithm. Thus, the key question is

how can the variable gains of position controllers generated by OGA reduce the contour error? Answering this question, it is essential to analyze the procedure in Fig. 2.6 as follows.

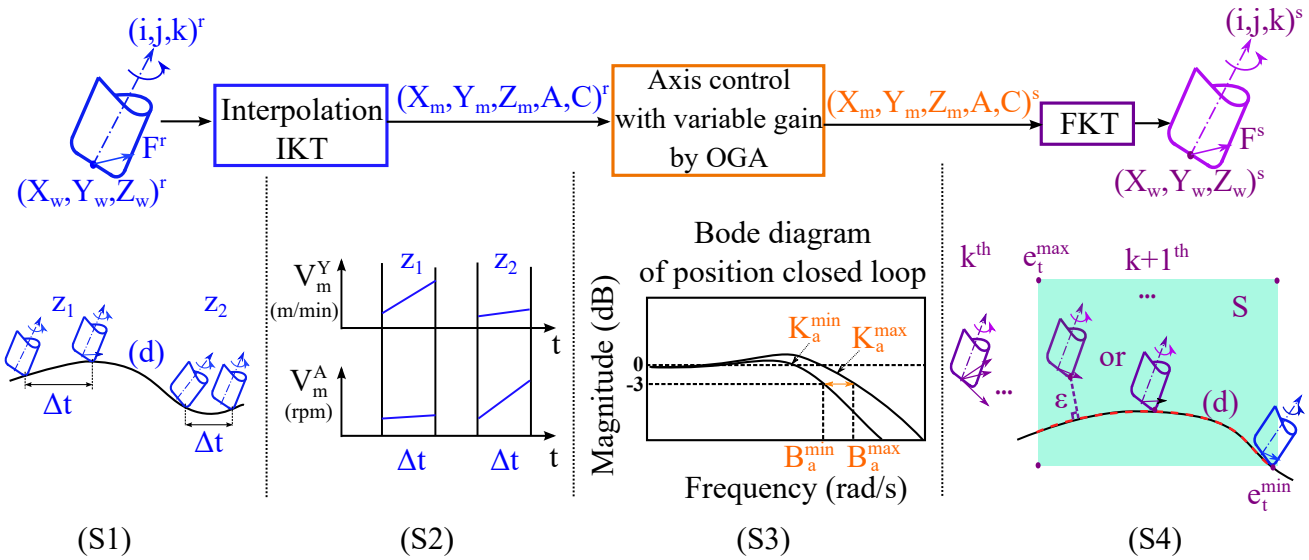

Figure 2.6: Motivation of gain adjustment

(S1) Along the desired profile (d) generated by CAM, different zones z have different curvature characteristics, e.g. z_1 and z_2 having small and high curvatures, respectively. The tool behaviors involve the tool location $\mathbf{C_L}(X_w, Y_w, Z_w)$ and the tool axis orientation $\mathbf{u}(i, j, k)$, while the tangential feedrate vector at the tool location point is denoted by $\mathbf{F}$. As discussed in Chapter 1, Section § 4, CAM only produces two inputs for the feedrate interpolation stage: the first one is the constant programmed feedrate and the second one is the programmed tool path.

(S2) According to the above desired tool behavior, the position setpoints $(X_m, Y_m, Z_m, A, C)^r$ generated by the feedrate interpolation stage have different kinematic characteristics between the different zones and between the different types of axis. For example: in the zone z_1, Y axis is more excited than A axis, while in z_2, the reverse is the case. These setpoints are then fed into the position closed loops of the axis drives to control the tool.

(S3) The axis tracking error depends partly on the reference dynamics (i.e spectral content). For axes with references having different dynamics, this error would be different for each axis, having an impact in the contour error. To reduce the contour error, one possibility is to modify the axis drive dynamics in order to adapt it in function of the reference of each axis.

If the controllers use fixed control gains, the dynamic of axis drives is kept unchanged during the

whole process of axis control. In the qualitative point of view, it can be said that the axis drive dynamic is too rigid to adapt appropriately the suitable positioning compliance. That causes the deviation between the resulting tool path and the reference one, that is responsible for the contour error presence.

Therefore, there is a need to excite more or less the axis drive dynamic during the axis control process to compensate the contour error. It can also be remarked that the axes are not needed to be permanently controlled with high gains, because for slow dynamics, a small gain is enough and the high frequencies of the machine are less excited. The OGA is derived based on such an idea.

Basically, when OGA adjusts the proportional gain of the position controller, K_a^P, it leads to adjusting of the dynamics of axis drives [Pritschow 1996], meaning that the bandwidth of the position loop B_a is modified within the range in (2.1),

$$B_a^{min} \leq B_a \leq B_a^{max} \tag{2.1}$$

when OGA adjusts the feed forward gain of the position controller, K_a^F, it modifies the impact of feed forward action on the velocity loop.

According to the above effects of the two control gain adjustments, denoting K_a as K_a^P and K_a^F, when OGA adjusts K_a within a boundary as in (2.2), it leads to the fact that the axis tracking error, $E_{a,m}$, is also modified within a boundary as in (2.3),

$$K_a^{min} \leq K_a \leq K_a^{max} \tag{2.2}$$

$$E_{a,m}^{min} \leq E_{a,m} \leq E_{a,m}^{max} \tag{2.3}$$

where $E_{X,m}$ is for example the axis tracking error of X axis, given in (2.4).

$$E_{X,m} = X_m^r - X_m^s \tag{2.4}$$

(S4) By solving the FKT, a set of tool behavior responses in the workpiece space is determined. Clearly, once the axis motions are adjusted more or less differently, the amplitude and direction of the resulting tangential feedrate vector in the workpiece space are also modified. This leads to the modification of the tool tracking error vector, e_t, which is defined in (1.5). Its norm is also bounded by (2.5).

$$e_t^{min} \leq \|e_t\| \leq e_t^{max} \tag{2.5}$$

Assume that between e_t^{min} and e_t^{max}, S represents a region where the tool can approach due to the effects of the above control gain adjustments. In this region, the desired profile (d) illustrates the tool contact positions corresponding to the zero contour error.

It can be said that the gain adjustments in OGA can finally modify the tangential feedrate vector so that the tool contact point can approach the desired profile (d), in order to remove or reduce the contour error.

3.2 OGA integrated control structure

The control structure with specificity of OGA is illustrated in Fig. 2.7.a and explained below.

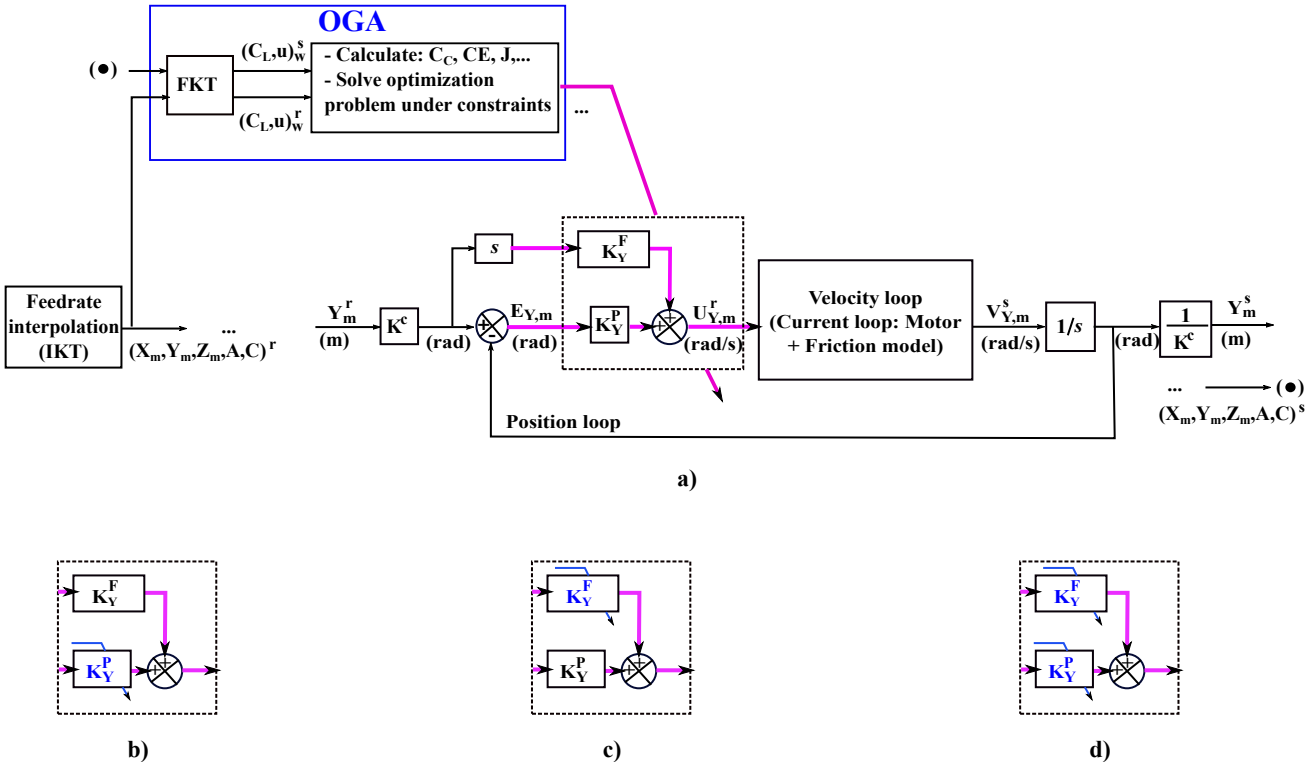

Figure 2.7: Control structure with OGA, e.g. for Y axis

In this figure, s and $1/s$ are the Laplace domain transfer functions for the derivative and integrator operations respectively; K^c is a conversion factor from m to rad . For simplicity, the velocity loop, current loop, motor and friction model have been hidden. The study intention is

to modify the control gains in position loop, K_a^P and K_a^F. The torque feed forward action as seen in Fig. 2.1 is not considered in this work, in order to highlight the effects of the velocity feed forward control action and its modification based on the OGA.

Note that the unit of K_a^P is 1/s. However, in [Prévost 2011a] K_a^P is proposed in m/min/mm for the linear axes (X, Y and Z) and in rad/min/mrad for the rotary axes (A and C). Both of them are multiplied a factor of 1000/60 to obtain the original unit 1/s. In the proceeding, K_a^P in either m/min/mm or rad/min/mrad will be used. While, K_a^F is a constant without unit.

The inputs of OGA are the position setpoints and the position responses simulated by the nonlinear axis model. In OGA, firstly the FKT problem is solved to obtain the reference and simulated tool behaviors in the workpiece space. Then the tool contact points and the contour error are estimated, based on the proposed estimation method in Chapter 1, Section § 3.1. The constraints are checked. Solving the contour error optimization problem under constraints allows one to obtain the optimal variable gains at the output of OGA. The gains are afterwards used to update the control gains in the position loop of axis drive.

In the following, three main cases of study of OGA are considered.

- Case 1: OGA optimizes only K_a^P, illustrated in Fig. 2.7.b.
- Case 2: OGA optimizes only K_a^F, illustrated in Fig. 2.7.c.
- Case 3: OGA optimizes both K_a^P and K_a^F, illustrated in Fig. 2.7.d.

3.3 Optimization problem of OGA

To pre-compensate the contour error before it happens in the online machining process, the OGA generates the optimal variable gains in the simulated machining framework by using the nonlinear axis model. This has been done based on the idea of Model Predictive Control (MPC), that is to formulate the optimization problem within the prediction horizon.

3.3.1 Horizon definition

At instant k, the future trajectory ranging from $k+1$ to $k+N^k$ within a horizon h^k is considered, as illustrated in Fig. 2.8.a. The horizon h^k is characterized by N^k instants or N^k tool positions, represented by $(\mathbf{C_L}, \mathbf{u})$. Its length is physically defined either as a function of time or as a function of tool displacement length, as proposed below.

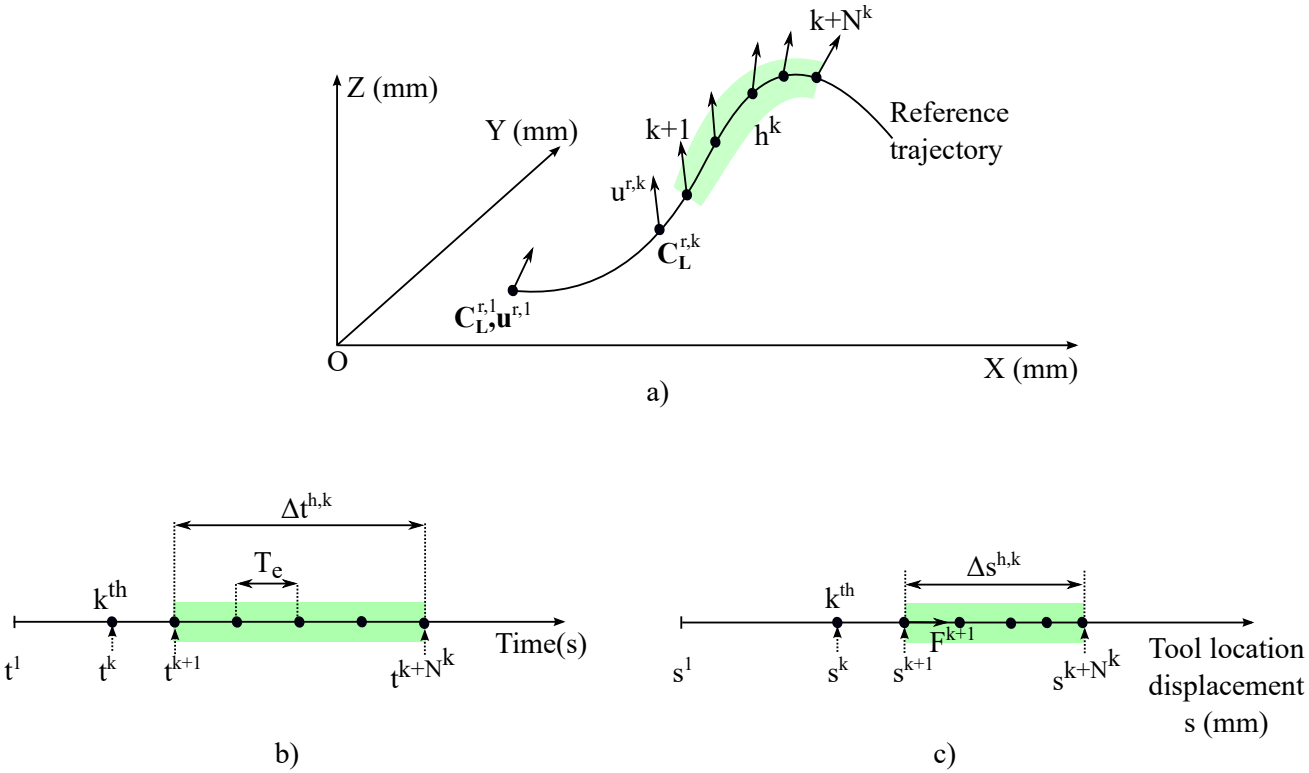

Figure 2.8: a) Horizon h^k; b) Temporal length of h^k; c) Geometrical length of h^k

In time domain, as illustrated in Fig. 2.8.b, the temporal length of h^k, $\Delta t^{h,k}$, is related to the sampling period of the position setpoint T_e and N^k through (2.6).

$$\Delta t^{h,k} = \left(N^k - 1\right) T_e \tag{2.6}$$

In the geometry domain, as illustrated in Fig. 2.8.c, the geometrical length of h^k, $\Delta s^{h,k}$ is defined by (2.7),

$$\Delta s^{h,k} = s^{k+N^k} - s^{k+1} \tag{2.7}$$

where s^k is defined in (1.11). In addition, $\Delta s^{h,k}$ can also be expressed through T_e and the

tangential feedrate F^i, as in (2.8).

$$\Delta s^{h,k} \cong T_e \left(\sum_{i=k+1}^{k+N^k-1} F^i \right) \tag{2.8}$$

If $\Delta t^{h,k}$ and $\Delta s^{h,k}$ are chosen in advance, N^k is obtained by solving (2.6) and (2.8), respectively.

There is a difference between the above two definitions of the horizon length. If all of the horizons over the trajectory are chosen as a function of time, having the same value of $\Delta t^{h,k}$, then each horizon has the same number of instants N^k due to the constant value of T_e. In contrast, if they are chosen as a function of tool displacement length, having the same value of $\Delta s^{h,k}$, then the number of instants N^k in each horizon may be different, depending on the different values of feedrate F^i. The two ways of the horizon length definition will be discussed in detail to solve the optimization of OGA in Chapter 3, Section § 2.1.

3.3.2 Optimization problem

Assume that at the k instant, the control gain values, the reference and simulated values of the axes motions and the tool behaviors at the present and past instants are known. The objective is to find out the optimal future values of the control gains from the instant $k+1$ to the instant $k+N^k$ so that the future contour errors are minimized, over the horizon h^k.

A matrix containing all position setpoints over the horizon h^k, $\mathbb{S}^r$, is defined in (2.9):

$$\mathbb{S}^r = \begin{bmatrix} X_m^{r,k+1} & Y_m^{r,k+1} & Z_m^{r,k+1} & A^{r,k+1} & C^{r,k+1} \\ X_m^{r,k+2} & Y_m^{r,k+2} & Z_m^{r,k+2} & A^{r,k+2} & C^{r,k+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ X_m^{r,k+N^k} & Y_m^{r,k+N^k} & Z_m^{r,k+N^k} & A^{r,k+N^k} & C^{r,k+N^k} \end{bmatrix} \tag{2.9}$$

Assume that within such a horizon, the values of K_a^P and K_a^F for all axes are given by the matrix $\mathbb{K}^P$ in (2.10) and $\mathbb{K}^F$ in (2.11), respectively.

$$\mathbb{K}^P = \begin{bmatrix} K_X^{P,k+1} & K_Y^{P,k+1} & K_Z^{P,k+1} & K_A^{P,k+1} & K_C^{P,k+1} \\ K_X^{P,k+2} & K_Y^{P,k+2} & K_Z^{P,k+2} & K_A^{P,k+2} & K_C^{P,k+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ K_X^{P,k+N^k} & K_Y^{P,k+N^k} & K_Z^{P,k+N^k} & K_A^{P,k+N^k} & K_C^{P,k+N^k} \end{bmatrix} \tag{2.10}$$

$$\mathbb{K}^F = \begin{bmatrix} K_X^{F,k+1} & K_Y^{F,k+1} & K_Z^{F,k+1} & K_A^{F,k+1} & K_C^{F,k+1} \\ K_X^{F,k+2} & K_Y^{F,k+2} & K_Z^{F,k+2} & K_A^{F,k+2} & K_C^{F,k+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ K_X^{F,k+N^k} & K_Y^{F,k+N^k} & K_Z^{F,k+N^k} & K_A^{F,k+N^k} & K_C^{F,k+N^k} \end{bmatrix} \tag{2.11}$$

The instant variation amounts of the above gain values of the axis a are denoted as δK_a^P and δK_a^F, and defined in (2.12) and (2.13), respectively. Note that $K_a^{P,k}$ and $K_a^{F,k}$ are known at the k instant.

$$\delta K_a^P = \begin{bmatrix} K_a^{P,k+1} - K_a^{P,k} \\ K_a^{P,k+2} - K_a^{P,k+1} \\ \cdots \\ K_a^{P,k+N^k} - K_a^{P,k+N^k-1} \end{bmatrix} \tag{2.12}$$

$$\delta K_a^F = \begin{bmatrix} K_a^{F,k+1} - K_a^{F,k} \\ K_a^{F,k+2} - K_a^{F,k+1} \\ \cdots \\ K_a^{F,k+N^k} - K_a^{F,k+N^k-1} \end{bmatrix} \tag{2.13}$$

The sums of squares of the instant gain variations of K_a^P and K_a^F are denoted as ΔK_a^P and ΔK_a^F, and defined in (2.14) and (2.15), respectively.

$$\Delta K_a^P = \left(\delta K_a^P\right)^T \left(\delta K_a^P\right) \tag{2.14}$$

$$\Delta K_a^F = \left(\delta K_a^F\right)^T \left(\delta K_a^F\right) \tag{2.15}$$

The vectors containing the above gain variations for all of the axes, denoted $\Delta\mathbb{K}^P$ and $\Delta\mathbb{K}^F$, are defined in (2.16) and (2.17), respectively.

$$\Delta\mathbb{K}^P = \left[\Delta K_X^P, \Delta K_Y^P, \Delta K_Z^P, \Delta K_A^P, \Delta K_C^P\right] \tag{2.16}$$

$$\Delta\mathbb{K}^F = \left[\Delta K_X^F, \Delta K_Y^F, \Delta K_Z^F, \Delta K_A^F, \Delta K_C^F\right] \tag{2.17}$$

To simulate the machining from the $k+1$ to $k+N^k$ instant over the horizon h^k, each row of $\mathbb{S}^r$ in (2.9), $\mathbb{K}^P$ in (2.10) and $\mathbb{K}^F$ in (2.11) are simultaneously sent to the machine simulator to generate the predicted position responses, denoted by $\hat{\mathbb{S}}^s$, and given by (2.18).

$$\hat{\mathbb{S}}^s = \begin{bmatrix} \hat{X}_m^{s,k+1} & \hat{Y}_m^{s,k+1} & \hat{Z}_m^{s,k+1} & \hat{A}^{s,k+1} & \hat{C}^{s,k+1} \\ \hat{X}_m^{s,k+2} & \hat{Y}_m^{s,k+2} & \hat{Z}_m^{s,k+2} & \hat{A}^{s,k+2} & \hat{C}^{s,k+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \hat{X}_m^{s,k+N^k} & \hat{Y}_m^{s,k+N^k} & \hat{Z}_m^{s,k+N^k} & \hat{A}^{s,k+N^k} & \hat{C}^{s,k+N^k} \end{bmatrix} \tag{2.18}$$

From (2.18), solving the FKT problem allows one to obtain a vector representing the predicted tool behaviors, $\hat{\mathbb{P}}^s$, given in (2.19),

$$\hat{\mathbb{P}}^s = \left[\hat{P}_w^{s,k+1}, \hat{P}_w^{s,k+2}, ..., \hat{P}_w^{s,k+N^k}\right] \tag{2.19}$$

where $\hat{P}_w^{s,k+1}$ represents for $(\hat{\mathbf{C}}_{\mathbf{L}}^{s,k+1}, \hat{\mathbf{u}}^{s,k+1})$, as illustrated in Fig. 2.9.a.

From (1.1) to (1.17) and (2.19), a vector containing the predicted tool contact points, $\hat{\mathbb{C}}_C^s$, is obtained in (2.20).

$$\hat{\mathbb{C}}_C^s = \left[\hat{\mathbf{C}}_{\mathbf{C}}^{s,k+1}, \hat{\mathbf{C}}_{\mathbf{C}}^{s,k+2}, ..., \hat{\mathbf{C}}_{\mathbf{C}}^{s,k+N^k}\right] \tag{2.20}$$

As illustrated in Fig. 2.9.b, the contour error is calculated by the orthogonal distance from each element of $\hat{\mathbb{C}}_C^s$ to the reference $\mathbf{C_C}$ path. To do this, as discussed in Chapter 1, Section § 3.1.3, a vector containing a set of the reference tool contact points, denoted as $\mathbb{C}_C^r$, is employed.

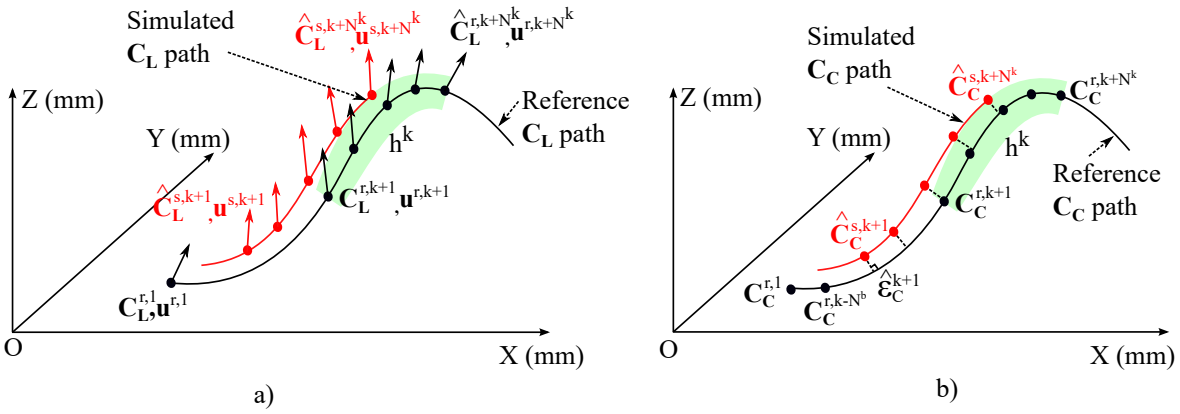

Figure 2.9: a) Predicted tool behaviors; b) Predicted contact points and predicted contour error

As it can be noticed, all of the simulated tool contact points from $\hat{\mathbf{C}}_{\mathbf{C}}^{s,k+1}$ to $\hat{\mathbf{C}}_{\mathbf{C}}^{s,k+N^k}$ lag behind their reference ones, from $\mathbf{C}_{\mathbf{C}}^{r,k+1}$ to $\mathbf{C}_{\mathbf{C}}^{r,k+N^k}$, due to the inherent tool tracking error. Therefore, the size of $\mathbb{C}_C^r$ must be greater than that of $\hat{\mathbb{C}}_C^s$ to ensure that the contour error can always be calculated. Consequently, $\mathbb{C}_C^r$ is proposed by (2.21),

$$\mathbb{C}_C^r = \left[\mathbf{C}_{\mathbf{C}}^{r,k-N^b}, \mathbf{C}_{\mathbf{C}}^{r,k-N^b+1}, ..., \mathbf{C}_{\mathbf{C}}^{r,k+N^k}\right] \tag{2.21}$$

in which $0 \leq N^b \leq k-1$ is chosen in advance so that $s^{r,k-N^b} < s^{s,k}$, where s^k defined in (1.11). If $N^b = k-1$, the first element of vector $\mathbb{C}_C^r$ is always $\mathbf{C}_{\mathbf{C}}^{r,1}$, the contour error can always be calculated, however it induces a computation burden. This is because the size of $\mathbb{C}_C^r$ will be

increased when k increases. To overcome such an inconvenience, depending on the dynamic response of the axis drive, N^b can be chosen as a specific value to reduce the size of $\mathbb{C}_C^r$ and so reduce the computation time. For example, during simulations of OGA with the axis drive in this thesis, N^b is chosen as in (2.22).

$$\begin{cases} N^b = k-1 & \text{if} \quad 1 \leq k \leq 100 \\ N^b = 100 & \text{if} \quad k > 100 \end{cases} \tag{2.22}$$

From (1.18) to (1.20), (2.20) and (2.21), the predicted contour error values over the horizon h^k are calculated and represented by $\hat{\mathbb{E}}_C$ in (2.23).

$$\hat{\mathbb{E}}_C = \left[\hat{\varepsilon}_C^{k+1}, \hat{\varepsilon}_C^{k+2}, ..., \hat{\varepsilon}_C^{k+N^k}\right] \tag{2.23}$$

From (2.16), (2.17), and (2.23), the cost function, J, is proposed in (2.24),

$$J = \hat{\mathbb{E}}_C \left(\hat{\mathbb{E}}_C\right)^T + \lambda^P \left(\Delta\mathbb{K}^P\right)^T + \lambda^F \left(\Delta\mathbb{K}^F\right)^T \tag{2.24}$$

where λ^P, λ^F are the weighting vectors for the gain variations of K_a^P and K_a^F, given in (2.25) and (2.26) respectively.

$$\lambda^P = \left[\lambda_X^P, \lambda_Y^P, \lambda_Z^P, \lambda_A^P, \lambda_C^P\right] \tag{2.25}$$

$$\lambda^F = \left[\lambda_X^F, \lambda_Y^F, \lambda_Z^F, \lambda_A^F, \lambda_C^F\right] \tag{2.26}$$

From (2.24), the optimization problem of OGA is formulated by (2.27),

$$\left[\tilde{\mathbb{K}}^P, \tilde{\mathbb{K}}^F\right] = argmin\,(J) \tag{2.27}$$

where $\tilde{\mathbb{K}}^P$ and $\tilde{\mathbb{K}}^F$ are the optimal values of $\mathbb{K}^P$ in (2.10) and $\mathbb{K}^F$ in (2.11).

In solving (2.27), the meaning of the weighting factors is that smaller λ^P and λ^F, greater $\Delta\mathbb{K}_a^P$ and $\Delta\mathbb{K}_a^F$ can be accepted, and vice-versa. The optimization problem in (2.27) is subject to constraints, that are discussed below.

3.4 Constraints

In order to assure machine kinematic limitations, stability and feasibility, the optimization of OGA in (2.27) is subject to the three following constraints.

3.4.1 Stability criterion

The first constraint is the stability criterion of the servo drive. It is proposed that the variable values of proportional gain in the position controller, generated by OGA, need to respect the limitations of phase margin and gain margin of the position open loop of the axis drive, as in (2.28) [Younkin 2003].

$$\begin{cases} Gain\ margin\ \geq 15\ \ (dB) \\ Phase\ margin\ \geq 70\ \ (Deg) \end{cases} \tag{2.28}$$

3.4.2 Axis kinematic limitations

The second constraint refers to the axis kinematic limitations. They are the axis velocity, acceleration and jerk limitations, given in Table 2.2, denoted by V_a^{max}, A_a^{max}, and J_a^{max} respectively, and represented in (2.29).

Table 2.2: Axis kinematic constraints of Mikron UCP 710 machine

	X	Y	Z	A	C
$V^{max}(m/min - rpm)$	30	30	30	15	20
$A^{max}(m/s^2 - rad/s^2)$	2.5	3	2.1	$0.83 \times 2\pi$	$0.83 \times 2\pi$
$J^{max}(m/s^3 - rad/s^3)$	5	5	50	$5 \times 2\pi$	$100 \times 2\pi$

$$\begin{cases} -V_a^{max} \leq \hat{V}_a^{s,i} \leq V_a^{max} \\ -A_a^{max} \leq \hat{A}_a^{s,i} \leq A_a^{max} \\ -J_a^{max} \leq \hat{J}_a^{s,i} \leq J_a^{max} \end{cases} \quad \text{with} \quad k+1 \leq i \leq k+N^k \tag{2.29}$$

where $\hat{V}_a^{s,i}$ for example are the predicted velocity values of the axis a from $k+1$ to $k+N^k$ instant. $\hat{V}_X^{s,k+1}$, $\hat{A}_X^{s,k+1}$ and $\hat{J}_X^{s,k+1}$ of X axis for example are calculated by (2.30).

$$\begin{cases} \hat{V}_X^{s,k+1} & = \dfrac{\hat{X}_m^{s,k+1} - X_m^{s,k}}{T_e} \\ \hat{A}_X^{s,k+1} & = \dfrac{\hat{V}_X^{s,k+1} - V_X^{s,k}}{T_e} \\ & = \dfrac{\hat{X}^{s,k+1} - 2X^{s,k} + X^{s,k-1}}{T_e^2} \\ \hat{J}_X^{s,k+1} & = \dfrac{\hat{A}_X^{s,k+1} - A_X^{s,k}}{T_e} \\ & = \dfrac{\hat{X}^{s,k+1} - 3X^{s,k} + 3X^{s,k-1} - X^{s,k-2}}{T_e^3} \end{cases} \tag{2.30}$$

3.4.3 Motor current limitations

Relating to the motor current response, it is noted that the sampling time of the position loop T_e is different from that of the current loop T_e'. The k instant of the position response corresponds to the $k^{'}$ instant of the current response. The relation between T_e, T_e', k and $k^{'}$ is evaluated through the machining time Δt_m, as represented in (2.31).

$$\Delta t_m = (k-1)\, T_e = \left(k^{'} - 1\right) T_e^{'} \tag{2.31}$$

There are two important values related to the motor current. The first one is the instant value of motor current $I_a^{k^{'}}$ (A). The second one is its increment $\Delta I_i^{k^{'}}$ (A), which is defined in (2.32).

$$\Delta I_a^{k^{'}} = I_a^{k^{'}} - I_a^{k^{'}-1} \tag{2.32}$$

The two values are subject to their constraints in (2.33),

$$\begin{cases} I_a^{k_i^{'}} \leq I_a^{k_i^{'},max} \\ \left|\Delta I_a^{k_i^{'}}\right| \leq \Delta I_a^{k_i^{'},max} \end{cases} \tag{2.33}$$

in which given $k^{'}$ in the prediction horizon h^k, $\Delta t_m\,(k+1) \leq \Delta t_m\left(k_i^{'}\right) \leq \Delta t_m\left(k+N^k\right)$; $I_a^{k_i^{'},max}$ is proportional to the admissible maximum of the motor torque $C_m^{max}\,(N.m)$ by a factor of $K_t\,(N.m/A)$, as in (2.34).

$$I_a^{k^{'},max} = C_m^{max}/K_t \tag{2.34}$$

Depending on the type of motor, e.g. the Siemens motor 1FT6084 - 8WF71 for Mikron Machine [Seimens 2005], the motor torque limit, C_m^{max}, is a function of the motor angular velocity, $V_m\,(rpm)$, through the "torque-velocity" diagram in Fig. 2.10, that is represented in (2.35).

$$C_m^{max} = f\,(V_m) \tag{2.35}$$

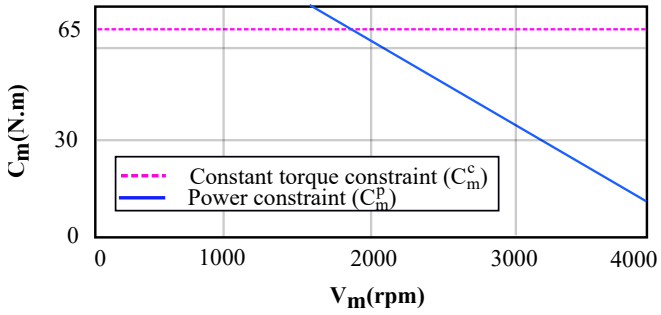

Figure 2.10: "Torque-Velocity" diagram of Siemens motor 1FT6084-8WF71

Therefore, from (2.34) and (2.35), the motor current limit is a function of K_t and V_m, as in (2.36).

$$I_a^{k',max} = f\,(V_m)\,/K_t \tag{2.36}$$

This means that during the control operation, each motor drive has an angular velocity profile, leading to a corresponding profile of the motor current limit $I_a^{k',max}$, as illustrated in Fig. 2.11 (black dashed curve).

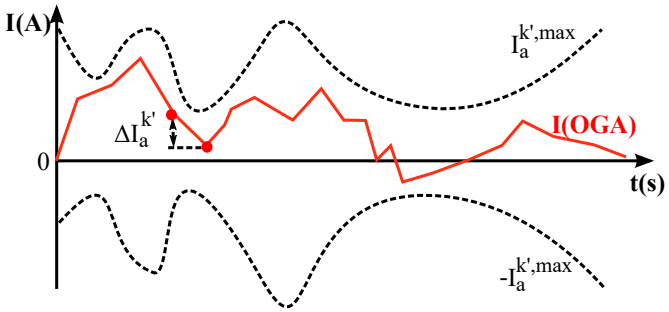

Figure 2.11: Motor current response of OGA

In addition, respecting the limit profile of motor current is not sufficient if the response of the motor current is too noisy and contains a large variation, because this fact can potentially

damage the motor and further the machine. Thus, it is proposed that the maximum current increment is equal to k_I (%) of the nominal value of motor current, I_a^n, which corresponds to the value of the constant torque constraint, C_m^c, in Fig. 2.10.

Consequently, I_a^n and $\Delta I_a^{k',max}$ are given in (2.37) and (2.38), respectively.

$$I_a^n = C_m^c / K_t \tag{2.37}$$

$$\Delta I_a^{k',max} = k_I I_a^n \tag{2.38}$$

To summarize, at the k instant the OGA aims at solving the optimization problem given in (2.27), being subject to the constraints formulated by (2.28), (2.29), and (2.33), to generate a set of optimal variable control gains used in the position controller of one of the three cases of study, illustrated in Fig. 2.7.b, c and d, in pre-compensating the contour error predicted by (2.23). The developments of OGA are valid for both the temporal horizon length in (2.6) and the geometrical horizon length in (2.7).

3.5 Admissible gain range

For each variable gain value at each instant, OGA always checks its feasibility based on the three above constraints. If one certain gain value does not satisfy the constraints, it will not be chosen. Therefore, the control gain range, in which OGA can be performed, can be theoretically unlimited. However, it is clear that if OGA is performed within an unlimited gain ranges, the computation burden turns to be too serious to finish the OGA optimization.

To address this issue, OGA should be performed within one limited range for each control gain, so-called admissible gain range. It is proposed to define the admissible gain range in the way that the classical axis control can use any fixed gain values in this range without violating any constraints in § 3.4.

For the three cases of study illustrated in Fig. 2.7.b, c and d, the admissible gain ranges of both K_a^P and K_a^F are established as follows.

- For the admissible range of K_a^P:
  The upper limit of K_a^P, $K_a^{P,max}$, is chosen respecting the stability criterion of the position

loop in (2.28), expressed in (2.39).

$$K_a^P \leq K_a^{P,max} \quad s.t \quad \text{stability criterion in (2.28)} \tag{2.39}$$

The lower limit of K_a^P, $K_a^{P,min}$, is globally proposed as in (2.40).

$$0 < K_a^{P,min} < K_a^{P,max} \tag{2.40}$$

One specific value of $K_a^{P,min}$ will be chosen by users.

- For the admissible range of K_a^F:
  The lower limit of K_a^F, $K_a^{F,min}$, should be zero, that is the case having only P controller activated, and is described by (2.41).

$$K_a^{F,min} = 0 \tag{2.41}$$

In fact, K_a^F is outside the position closed loop, its maximum value does not affect the stability criterion of this loop. However, the variation of K_a^F affects the behavior of the reference velocity signal in the velocity loop. A too large value of K_a^F can violate the axis kinematic limitations and/or the current limitations.

The upper limit of K_a^F, $K_a^{F,max}$, depends on two following factors. The first one is the reference axis kinematic profiles, including the reference axis velocity, acceleration and jerk profiles. One example for the reference jerk profile is that, as it can be seen in Fig. 2.12, case 1 has the reference jerk profile much below its limitations than case 2. Thus, there are more spaces for the variations of the actual axis jerks in case 1 than in case 2. Consequently, the potential $K_a^{F,max}$ in the former can be larger than in the later, $K_a^{F,max,1} > K_a^{F,max,2}$.

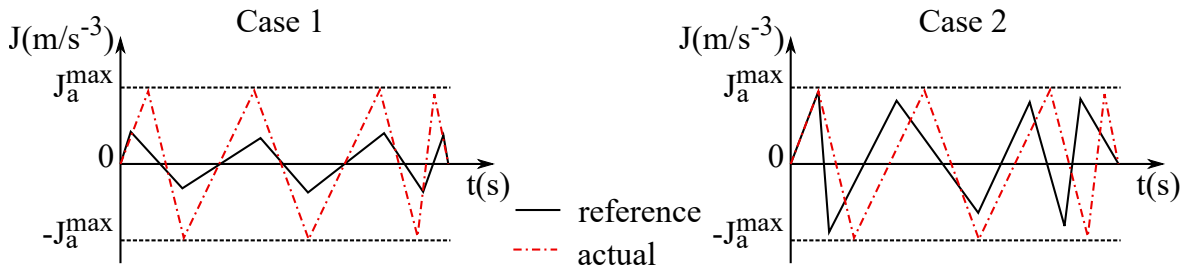

Figure 2.12: Reference and actual axis jerks

The second factor is the chosen value of K_a^P. For example, as it can be seen in Fig. 2.13, the reference velocity signal $U_{a,m}^r$ is a sum of two control signals U_a^P and U_a^F, derived from

K_a^P and K_a^F respectively. Obviously, when $U_{a,m}^r$ is subject to a limit value, the chosen K_a^P value will affect the upper limit of K_a^F, and vice versa.

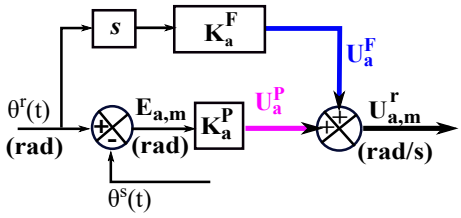

Figure 2.13: Impacts of K_a^P and K_a^F on the reference velocity signal

In addition, the choice of $K_a^{F,max}$ is subject to the two constraints, including the axis kinematic limitations in (2.29) and the current limitations in (2.33).

According to the above facts related to $K_a^{F,max}$, it is proposed two steps for obtaining its value as follows.

– Firstly, the admissible gain range of K_a^P is discretized into numerous discrete values. With each discrete value of K_a^P, increasing the value of K_a^F from zero till one limit value, over which the axis kinematic limitations and the current limitations are not satisfied anymore. This first step is represented by (2.42),

$$\begin{cases} K_a^P = K_a^{P,1}, \quad K_a^F \leq K_a^{F,1} \\ K_a^P = K_a^{P,2}, \quad K_a^F \leq K_a^{F,2} \\ \dots \\ K_a^P = K_a^{P,N}, \quad K_a^F \leq K_a^{F,N} \end{cases} \quad s.t. \quad \begin{cases} \text{axis kinematic limitations in (2.29)} \\ \text{current limitations in (2.33)} \end{cases} \tag{2.42}$$

with $1 \leq i \leq N$, $K_a^{P,min} \leq K_a^{P,i} \leq K_a^{P,max}$.

– Secondly, obtain $K_a^{F,max}$ by (2.43).

$$K_a^{F,max} = \min\left(K_a^{F,1}, K_a^{F,2}, \dots, K_a^{F,N}\right) \tag{2.43}$$

To summarize, the control gains $\mathbb{K}^P$ in (2.10) and $\mathbb{K}^F$ in (2.11) are bounded within the admissible gain ranges, represented in (2.44),

$$\begin{cases} K_a^{P,min} \leq K_a^{P,i} \leq K_a^{P,max} \\ K_a^{F,min} \leq K_a^{F,i} \leq K_a^{F,max} \end{cases} \quad \text{with} \quad k+1 \leq i \leq k+N^k \tag{2.44}$$

in which it can be highlighted that $K_a^{P,min}$ is chosen by users, respecting (2.40); $K_a^{P,max}$ is determined depending on the chosen stability criterion for the servo drive, as in (2.39); $K_a^{F,min}$ is obviously zero, as in (2.41); and $K_a^{F,max}$ is determined depending on the reference trajectory and the chosen value of K_a^P, being subject to the two constraints, as expressed in (2.43).

3.6 Solve the optimization problem

To solve the OGA optimization problem, two questions need to be answered. How to optimize the variable gains for the future trajectory over the horizon h^k in (2.10) and (2.11)? Which technique allows to solve the optimization problem in (2.27)? Following, two solving techniques are proposed to solve the optimization problem of OGA for the whole trajectory.

At the first instant, $k = 1$, the initial values of control gains are freely chosen respecting the admissible gain ranges in § 3.5 to start the simulated machining, as in (2.45).

$$\begin{cases} K_a^{P,1} = K_a^{P,initial} \\ K_a^{F,1} = K_a^{F,initial} \end{cases} \tag{2.45}$$

This leads to the fact that the first simulated/reference values of the axes motions and tool behaviors are achieved. Then, the prediction horizon starts at $k = 1$ and the OGA is solved to obtain optimal variable gains for $k > 1$.

3.6.1 First solution technique: OGA(a)

The first solution technique of OGA is denoted as OGA(a). Its motivation is illustrated in Fig. 2.14 and explained as follows.

Over the horizon h^k in Fig. 2.14.a, a gain modification profile is defined to generate variable values for the control gain K_a of the axis a, which are the gain values over the columns of either $\mathbb{K}^P$ in (2.10) or $\mathbb{K}^F$ in (2.11), as illustrated in Fig. 2.14.b. This gain profile is a function of several parameters. The principle of OGA(a) is to tune the parameter values of such a gain profile. By this way, different parameter values generate different responses of the gain profile, yielding different values of the control gains. Then, by evaluating the optimization problem in (2.27), the best parameter value of the gain function is chosen, corresponding to the optimal variable gains

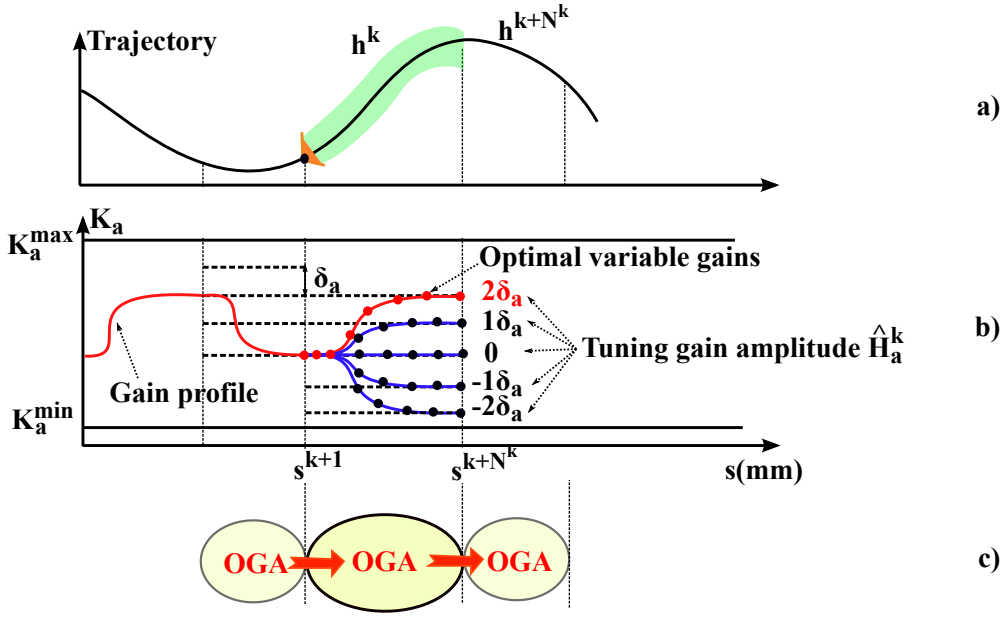

Figure 2.14: OGA(a): a) Horizon length; b) Tuning cases; c) Receding horizon without overlap

over the horizon h^k. These optimal variable gains are all kept and applied to control the tool position and orientation over the future trajectory within the concerned horizon. The horizon is then receded to the next one without overlap and the optimizing procedure is repeated, as illustrated in Fig. 2.14.c. The proposed gain function is illustrated in Fig. 2.15 and described below.

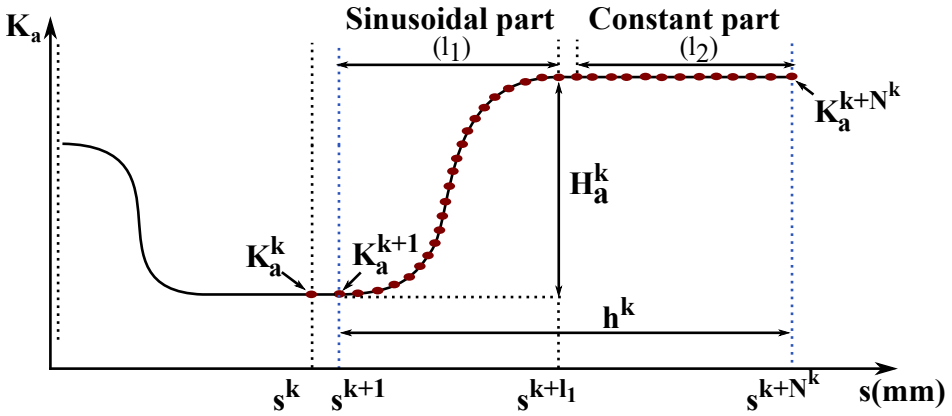

Figure 2.15: Gain prediction by predefined gain function

It consists of a sinusoidal part and a constant part, formed by (2.46) and (2.47), respectively,

$$(l_1): \; K_a^{k+1:k+l_1} = -\frac{1}{2}H_a^k \sin(t_s) + \frac{1}{2}H_a^k + K_a^k \tag{2.46}$$

$$(l_2): \; K_a^{k+l_1+1:k+N^k} = K_a^{k+l_1} \tag{2.47}$$

where

- The horizon length characterized by N^k can be a fixed value or adaptive values along the trajectory, depending on user's choice for either the temporal horizon length in (2.6) or the geometrical horizon length in (2.7).
- l_1 is the length of the sinusoidal part. This is an user-defined value, proposed in (2.48).

$$l_1 = k_l N^k \quad \text{with} \quad 0 < k_l \leq 1 \tag{2.48}$$

  If $k_l = 1$, the constant part defined in (2.47) is removed, there is only the sinusoidal part in (2.46). The meaning of l_1 is that it allows to change the shape of the gain profile generated by the proposed gain function, as illustrated in three examples in Fig. 2.16.

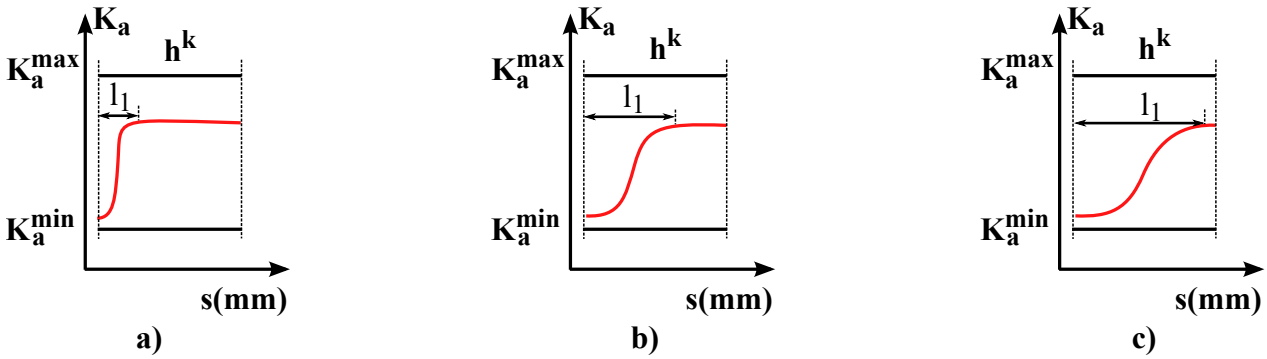

Figure 2.16: Effects of different values of l_1

- t_s is a time array of the sinusoidal part, defined by (2.49).

$$t_s = [\pi/2 : 1/(l_1 - 1) : 3\pi/2] \tag{2.49}$$

- H_a^k represents the gain amplitude modification of such a gain function. It can have a positive, zero or negative value. When the value of H_a^k is changed, different behaviors of the gain function are obtained for tuning, as illustrated in Fig. 2.14.b.

  In respecting the admissible gain range in (2.44), the constraint of H_a^k is given in (2.50).

$$K_a^{min} - K_a^k \leq H_a^k \leq K_a^{max} - K_a^k \tag{2.50}$$

There are many possibilities to propose such a gain function, e.g. using the parabola function instead of the sinusoidal one. However, only the above gain function is chosen, in order to create smooth gain transition between the two consecutive horizons, by choosing an appropriate value of l_1 in (2.48). Other possibilities will be checked in future works.

To solve the optimization problem of OGA, a gridding approach is used. In this approach, l_1 is chosen constant in advance and kept fixed for all of the horizons, the criteria in (2.24) is evaluated for H_a^k equal to each element of the vector U_a^k in (2.51),

$$U_a^k = \begin{bmatrix} -m\delta_a & -(m-1)\,\delta_a & \dots & 0\delta_a & \dots & (n-1)\,\delta_a & n\delta_a \end{bmatrix}_{(1,m+n+1)} \tag{2.51}$$

where

- The element $\{0\delta_a = 0\}$ of U_a^k means that if $H_a^k = 0$, all of the variable gains in the horizon h^k, defined by (2.46) and (2.47), are equal to the current gain value K_a^k.
- δ_a is an amplitude step of the gain function in (2.46). It is subject to (2.52).

$$0 < \delta_a \leq K_a^{max} - K_a^{min} \tag{2.52}$$

- Denote p_0 as the total number of potential gain values for K_a. From (2.51), p_0 is defined in (2.53).

$$p_0 = m + n + 1 \quad \text{with} \quad 0 \leq \{n, m\} \leq p_0 - 1 \tag{2.53}$$

- The constraint in (2.50) leads to the condition in (2.54).

$$\begin{cases} 0 \leq n\delta_a \leq K_a^{max} - K_a^k \\ 0 \leq m\delta_a \leq K_a^k - K_a^{min} \end{cases} \tag{2.54}$$

It is proposed that the maximum gain variation, $(p_0 - 1)\,\delta_a$, is always less than or equal to the admissible gain range, represented in (2.55),

$$(p_0 - 1)\,\delta_a = T_0\left(K_a^{max} - K_a^{min}\right) \quad \text{with } 0 < T_0 \leq 1 \tag{2.55}$$

in which T_0, denoted for the variation factor, is an user-defined parameter. It allows to regulate the maximum gain variation. If $T_0 = 1$, the maximum gain variation is equal to the admissible

gain range. If $0 < T_0 < 1$, the maximum gain variation is smaller than the admissible gain range.

From (2.55), the gain amplitude step, δ_a, is defined in (2.56).

$$\delta_a = \frac{T_0 \left(K_a^{max} - K_a^{min}\right)}{(p_0 - 1)} \tag{2.56}$$

If p_0 is a fixed value, the value of T_0 also allows to adjust the amount of gain step.

To determine n and m in (2.51), and respect the constraint in (2.53), n is firstly proposed by (2.57) and illustrated in Fig. 2.17.

$$(q): \quad n = \frac{p_0 - 1}{K_a^{max} - K_a^{min}} \left(-K_a^k + K_a^{max}\right) \quad \text{with} \quad K_a^k \in \left[K_a^{min} : K_a^{max}\right] \tag{2.57}$$

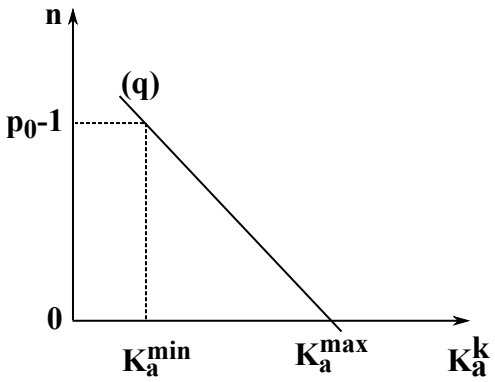

Figure 2.17: Value of n as a function of K_a^k

The meaning of (2.57) is that if $K_a^k = K_a^{min}$, then $n = p_0 - 1$. If $K_a^k = K_a^{max}$, then $n = 0$. Therefore, when K_a^k ranges from K_a^{min} to K_a^{max}, n ranges from $p_0 - 1$ to 0.

Afterwards, from (2.53), m is defined in (2.58).

$$m = p_0 - n - 1 \tag{2.58}$$

It can be easily verified that δ_a in (2.56), n in (2.57) and m in (2.58) respect the constraints in (2.54). To facilitate the calculation, n and m are chosen as integers. It is proposed that after solving (2.57), only the nearest integer less than or equal to the value of n is used. Then, m is calculated by (2.58). If $m\delta_a$ is not satisfied (2.54), meaning that $m\delta_a > K_a^k - K_a^{min}$, then $m\delta_a$ is set as $K_a^k - K_a^{min}$.

In the case of $T_0 = 1$, Fig. 2.18 shows three examples of the tuning cases of U_a^k in (2.51). Given $p_0 = 5$, from (2.56) $\delta_a = (K_a^{max} - K_a^{min})/4$.

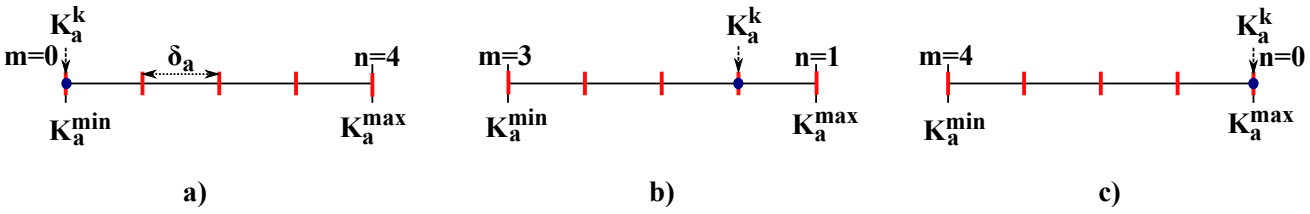

Figure 2.18: Three examples for the tuning case of $T_0 = 1$

- In Fig. 2.18.a, if $K_a^k = K_a^{min}$, from (2.57) and (2.58), leading $n = 4$ and $m = 0$. Thus, U_a^k in (2.51) is $[0\delta_a, 1\delta_a, 2\delta_a, 3\delta_a, 4\delta_a]$.
- In Fig. 2.18.b, if $K_a^k = K_a^{max} - \delta_a$, from (2.57) and (2.58), leading $n = 1$ and $m = 3$. Thus, U_a^k in (2.51) is $[-3\delta_a, -2\delta_a, -1\delta_a, 0\delta_a, 1\delta_a]$.
- In Fig. 2.18.c, if $K_a^k = K_a^{max}$, from (2.57) and (2.58), leading $n = 0$ and $m = 4$. Thus, U_a^k in (2.51) is $[-4\delta_a, -3\delta_a, -2\delta_a, -1\delta_a, 0\delta_a]$.

Similarly, in the case of $T_0 = 0.5$, Fig. 2.19 shows three examples of the tuning cases of U_a^k in (2.51).

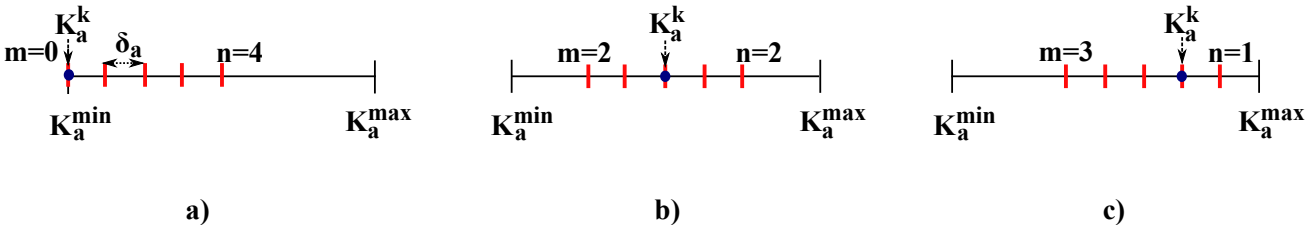

Figure 2.19: Three examples for the tuning case of $T_0 = 0.5$

It can be seen that in this case the tuning gain range, the maximum gain variation and the gain step are half of the ones in the previous case with $T_0 = 1$. From Fig. 2.19.a to Fig. 2.19.c, it can be noticed a receding effect of the tuning gain range, that does not exist when $T_0 = 1$.

To summarize, the OGA(a) has three main properties: (1) generate the variable gains using a predefined gain transition function as a template, (2) tune and optimize the gain amplitude of

the gain function within the admissible gain range to generate the optimal variable gains and (3) recede the horizon without overlap.

The gain modifications over the trajectory are defined by parameters N^k, k_l, T_0 and H_a^k. The parameters N^k, k_l and T_0 are fixed in advance and are equal for each axis. Only parameter H_a^k, one for each axis, is obtained by optimization.

3.6.2 Second solution technique: OGA(b)

The second solution technique of OGA is denoted as OGA(b). It takes the advantages of MPC into account, that is to find the optimal gains for the whole horizon h^k at the k instant but only the first optimal gain of the horizon is kept for the instant $k+1$, as illustrated in Fig. 2.20.a and b. Then the horizon is receded to the instant $k+1$ and the optimization is repeated, as shown in Fig. 2.20.c. The method of receding horizon allows one to apply the OGA optimization for the whole trajectory.

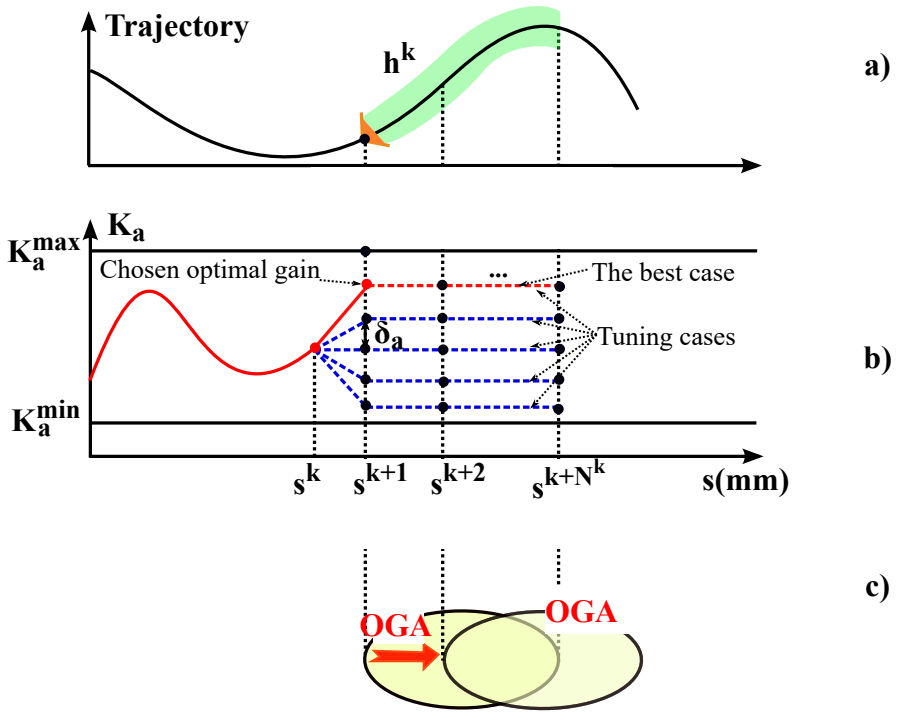

Figure 2.20: OGA(b): a) Horizon length; b) Tuning the constant gains; c) Receding horizon with overlap

During the prediction horizon h^k, the gain can take different values for each sampling time. In order to simplify the optimization problem and as receding horizon is used, it is proposed that the gain is constant during the prediction horizon.

Assume that δK_a (2.59), representing for either δK_a^P in (2.12) or δK_a^F in (2.13), is the gain modifications within the horizon h^k.

$$\delta K_a = \begin{bmatrix} K_a^{k+1} - K_a^k \\ K_a^{k+2} - K_a^{k+1} \\ \dots \\ K_a^{k+N^k} - K_a^{k+N^k-1} \end{bmatrix} \tag{2.59}$$

Solving the OGA optimization problem with this method consists in tuning δK_a in (2.59) within the admissible gain ranges in (2.44). Applying a gridding approach to solve this optimization problem, the criteria (2.24) is evaluated for δK_a in (2.59) equal to the columns of matrix W_a in (2.60), to find out all of the possibilities of $\mathbb{K}^P$ in (2.10) and $\mathbb{K}^F$ in (2.11),

$$W_a^k = \begin{bmatrix} -m\delta_a & -(m-1)\,\delta_a & \dots & 0\delta_a & \dots & (n-1)\,\delta_a & n\delta_a \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 \\ : & : & : & : & : & : & : \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 \end{bmatrix}_{\left(N^k, m+n+1\right)} \tag{2.60}$$

in which as same as OGA(a), it is proposed to introduce the total number of potential gain values p_0 in (2.53), and the variation factor T_0 in (2.55). Then, the gain step δ_a, the values of n and m are defined in (2.56), (2.57) and (2.58), respectively.

One example is that if δK_a in (2.59) is equal to the last column of W_a^k in (2.60), the variable gains become (2.61):

$$\begin{cases} K_a^{k+1} = K_a^k + n\delta_a \\ K_a^{k+2:k+N^k} = K_a^{k+1} \end{cases} \tag{2.61}$$

Then, by checking all of the possibilities of $\mathbb{K}^P$ in (2.10) and $\mathbb{K}^F$ in (2.11) by means of the machine simulator, the optimal variable gains satisfying (2.27) are found.

The illustration for the values in the first row of W_a in (2.60) is similar to Fig. 2.18 and Fig. 2.19

4 Conclusion

To conclude, the main part of this chapter has been dedicated to the formulation of the optimization problem of OGA. The optimization problem of OGA is proposed in (2.27), being subject to the stability, axis kinematic, and current constraints, represented in (2.28), (2.29), and (2.33), respectively.

The optimization problem of OGA is basically built for the future trajectory over a horizon, in order to deal with the challenges in machining the free-form geometry. After solving this optimization problem, the horizon is receded so that the OGA optimization can cover the whole trajectory. Finally, an optimal variable gain profile is obtained, pre-compensating the contour error.

For solving such an optimization problem of OGA, there are many techniques that can be performed, such as genetic algorithm, heuristic method, etc. However, the two proposed solving methods for the OGA optimization problem have been preferred here, based on a gridding technique. The primary reason for this choice is its simplicity. That makes the OGA more practical and easily implementable in the machining community. However, such a technique also causes a problem related to computation burden. This drawback will be discussed in Chapter 3, Section § 2.2.

Chapter 3

Simulations and discussions

Contents

The optimization problem under constraints of OGA and the two solving techniques of OGA have been proposed in the previous chapter. The aim of this chapter is to analyze the efficiency of the above developments, especially in the contouring accuracy improvement and the constraint verification.

To do this, the necessary starting points and parameters configurations are performed. Many cases of study are carried out. Simulation results are shown and discussed to prove the performance of the proposed approach.

1 Starting points

Before performing and evaluating the OGA, the following tasks are considered:

- Generate the reference trajectory and position setpoints.
- Find out the admissible gain range, based on the analyses in Chapter 2, Section § 3.5.
- Find out the best responses of the classical axis control, consider it as the reference case to compare with the performance of OGA.
- Analyze the sensitivity of K^P and K^F in the position loop.

1.1 Reference trajectory and position setpoints

Fig. 3.1.a shows the desired part having free-form profile for milling with 5-axis Mikron UCP 710 machine center.

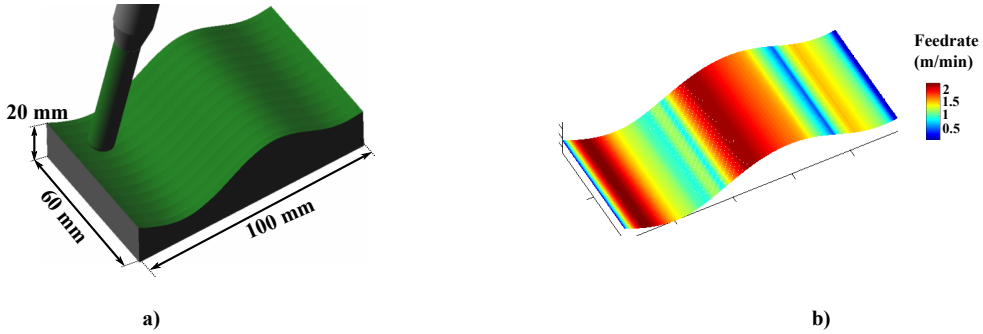

Figure 3.1: a) Desired part; b) Feedrate cartography from feedrate planning

The programmed tool path is created by CAM, with the configuration given in Appendix B.

The position setpoints are generated by the feedrate interpolation stage using the VPOp algorithm [Beudaert 2012], with the configuration of Bspline interpolation, the interpolation period of 6 ms and the programmed feedrate of 3 m/min. Without loss of generality in 5-axis machining and for the simplicity purpose, this interpolation task is carried out with only two linear axes, Y and Z, and one rotary axis, A. The resulting position setpoints correspond to the reference feedrate cartography illustrated in Fig. 3.1.b.

1.2 Admissible gain range

An important starting point of OGA is the admissible gain range. Based on the analysis in Chapter 2, Section § 3.5, the upper limit of K_a^P in the position controller should firstly be determined by respecting the stability criterion. To do so, the linear axis model of the position open loop should firstly be identified from the nonlinear one, that comes from the complete axis model in Fig. 2.7.a and simply illustrated in Fig. 3.2. The identification is based on a least square identification technique.

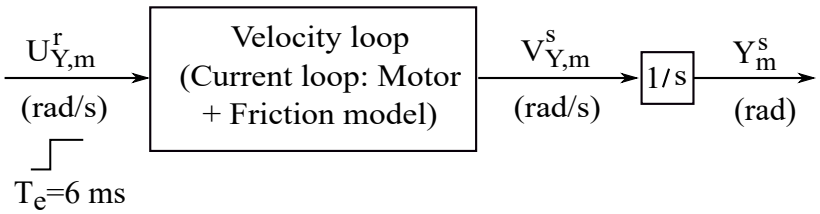

Figure 3.2: Nonlinear model of the position open loop without control gain ($T_e = 6$ ms)

To do such an identification, there are two steps: (1) identify the linear model of the velocity closed loop and (2) combine the obtained linear velocity model with the linear integrator action.

Firstly, the nonlinear velocity model is excited by a step signal. The velocity step response of the nonlinear model depends on the amplitude of the step, principally because of the friction in the machine. The identification has been done using a step amplitude equal to the half of the maximum velocity of the axes, that is because the nonlinear axis model is previously identified through this value as well.

Indeed, the numerical model needs to provide the same response as the nonlinear one. This

precision depends on the chosen order of the transfer function. Thus, an appropriate choice for the order of the model should be achieved.

Fig. 3.3 shows the comparison between the step velocity responses of the different order linear models and that of the nonlinear one, for example in the case of Y axis.

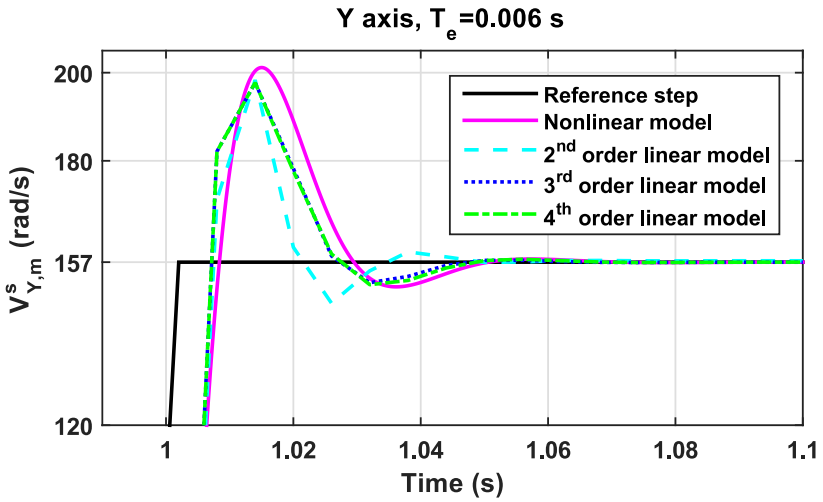

Figure 3.3: Step response of different order linear models of velocity loop

As it can be seen, the third and fourth order linear models approximate reasonably well the response of the nonlinear velocity loop of Y axis. In the following, the third order model is finally chosen as it is simpler than the fourth one and has the same response. Its mathematical representation in discrete time is given by (3.1),

$$\frac{V^s_{Y,m}\left(q^{-1}\right)}{U^r_{Y,m}\left(q^{-1}\right)} = \frac{1.16q^{-1} - 0.3656q^{-2}}{1 - 0.4007q^{-1} + 0.1411q^{-2} + 0.05416q^{-3}} \tag{3.1}$$

with q^{-1} the backward-shift operator.

Secondly, the discrete time transfer function of the integrator action is given in (3.2).

$$\frac{Y^s_m\left(q^{-1}\right)}{V^s_{Y,m}\left(q^{-1}\right)} = \frac{0.006}{1 - q^{-1}} \tag{3.2}$$

As a result, the linear model of the position open loop is determined by combining the linear velocity model in (3.1) with the integrator action (3.2), and given in (3.3).

$$\frac{Y^s_m\left(q^{-1}\right)}{U^r_{Y,m}\left(q^{-1}\right)} = \frac{0.00696q^{-1} - 0.002193q^{-2}}{1 - 1.401q^{-1} + 0.5418q^{-2} - 0.08693q^{-3} - 0.05416q^{-4}} \tag{3.3}$$

From the obtained linear model of the position open loop, the upper limit of K_Y^P can be obtained, by tuning this gain, illustrated in Fig. 3.4, respecting the stability criterion in (2.28).

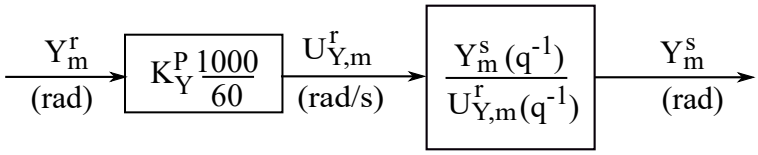

Figure 3.4: Position open loop with the control gain K_Y^P (m/min/mm) and its linear plan

The Black-Nichols chart of the position open loop of Y axis in Fig. 3.5 allows to compare the phase margin (dB) and gain margin (deg) for different K_Y^P values, in order to choose the allowable $K_Y^{P,max}$, satisfying (2.28).

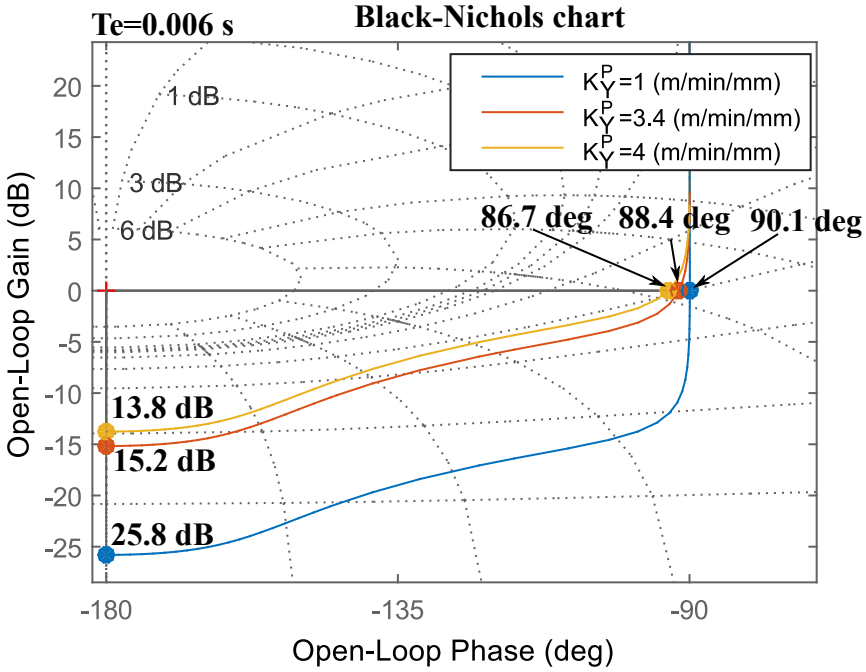

Figure 3.5: Frequency analysis for different K_Y^P of Y axis

In addition, the lower limit of K_a^P, $K_a^{P,min}$, is chosen in considering (2.40). Next, by considering (2.41) (2.42) and (2.43), the admissible range of K^F is obtained.

Consequently, the admissible ranges of K_a^P and K_a^F for OGA simulations are obtained and given in Table 3.1.

Table 3.1: Admissible gain range of K_a^P (m/min/mm or rad/min/mrad) and K_a^F

	Y	Z	A
K_a^P	[0.9 : 3.4]	[0.9 : 2.7]	[0.9 : 2.7]
K_a^F	[0 : 1.08]	[0 : 1.95]	[0 : 1.52]

Note that, the $K_a^{P,min}$ can be chosen as a very small positive value, satisfying (2.40). However, this increases the computation burden of OGA. To have a reasonable computation time for the simulations of OGA in many cases of study, only $K_a^{P,min} = 0.9$ is checked and shown.

1.3 Reference response in the classical axis control

To evaluate the benefit of variable gains obtained by OGA in axis control, the contouring accuracy of the classical axis control using fixed control gains is firstly discussed. Table 3.2 shows three cases of study using fixed control gains, in which the K^P values are the same as K^{P*} for the three cases, the K^F values are slightly different.

Table 3.2: Three cases of study for using fixed gains

Cases of study	Gains	Y	Z	A	$\bar{\varepsilon}$ (μm)
	K^{P*}(m/min/mm or rad/min/mrad)	1.2	1.8	1	
(1)	K^{F1}	0.95	0.85	0.95	33.3
(2)	K^{F2}	0.85	0.75	0.85	18.5
(3)	K^{F*}	0.9	0.8	0.9	14.4

Note that the third case, K^{P*} combined with K^{F*} are the best fixed gains, which are obtained as follows. Testing all combinations of K^P and K^F in the admissible range from Table 3.1, with a grid step of $\Delta K^P = 0.1$ for K^P and $\Delta K^F = 0.01$ for K^F, as illustrated by Fig. 3.6. Such values of ΔK^P and ΔK^F are chosen considering the sensitivity of these two gains on the reference velocity signal of the velocity loop, that will be discussed in the next section § 1.4. The smallest mean contour error have been obtained with K^{P*} and K^{F*}. This test takes about three days to finish with a PC Intel(R) Core(TM) i7-4790 CPU 3.60 GHz, RAM 16 Go and the testing algorithm is performed on the nonlinear axis model implemented in Matlab R2014b.

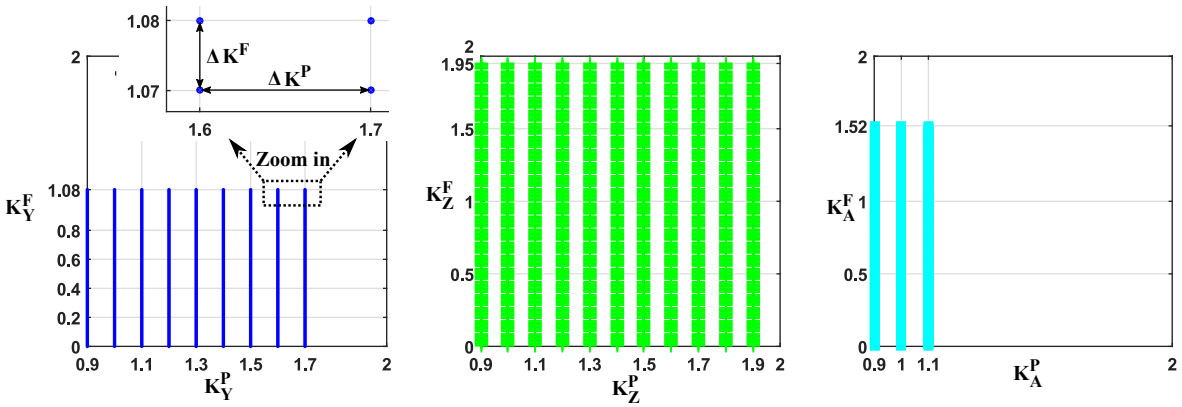

Figure 3.6: Discretized admissible gain range by the gain steps: $\Delta K^P = 0.1$ and $\Delta K^F = 0.01$

The contouring accuracy is quantitatively evaluated through the mean contour error, $\bar{\varepsilon}$, defined in (3.4),

$$\bar{\varepsilon} = \frac{1}{N} \sum_{i=1}^{N} \left| \varepsilon^i \right| \tag{3.4}$$

with N is the number of tool positionings on the trajectory.

As it can be seen from Table 3.2, even small changes in the K^F value can produce different mean contour error values.

Fig. 3.7 shows the curvature, feedrate (F) and contour error (CE) profiles as a function of the trajectory geometry for the above three cases of study.

Based on the curvature profile, the trajectory can be divided into nine portions, from S1 to S9, having the equivalent curvature characteristic in each portion. S2 and S8 are concave portions with the increasing curvatures. S3 and S9 are concave portions with the decreasing curvatures. S5 and S6 are convex portions having the increasing and decreasing curvatures respectively. S1, S4 and S7 have curvatures close to zero and can be seen as linear portions. In each portion, the feedrate can be increased and/or decreased, depending on the strategy in the feedrate interpolation task. As it can be seen, the feedrate in the three linear portions is increased quickly.

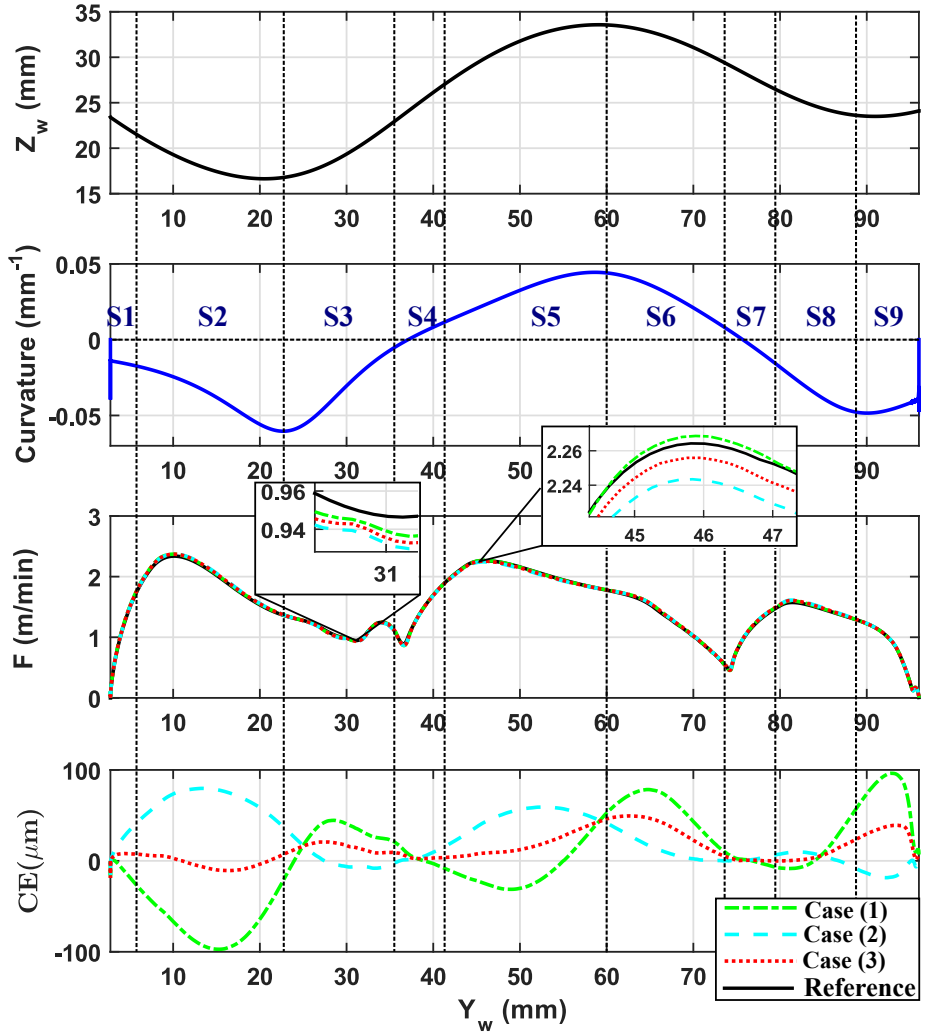

Figure 3.7: Responses with the fixed control gain values

As it can be observed, case (3) produces the smallest mean contour error in the three studied cases. The effects of different curvature portions and different gain values on the resulting contour error profiles are discussed as follows:

- The contour error in the high curvature portions, S2, S3, S5, S6, S8 and S9, is much bigger than in the linear portions, S1, S4 and S7.
- In the same curved portions, the increasing curvatures (S2, S5 and S8) and the decreasing curvatures (S3, S6 and S9) produce different contour error profiles.
- In S2, S5, and S6, there are slight differences in feedrate tracking, however, their corre-

sponding contour error profiles are greatly different. It means that the resulting contour error profile is very sensitive to the feedrate.

- One specified combination of the fixed gain values may produce a small contour error in some portions, but may cause a large contour error in other ones. For example, case (3) has smaller contour error than case (2) in S2 and S5; while case (2) has smaller contour error than case (3) in S3, S6 and S9.

From the above facts, there are three general remarks in using the fixed control gains, as follows:

- Firstly, large contour error usually occurs in the curved portions (S2, S5, S6, S9). Meanwhile, the linear portions usually have small contour error (S1, S4, S7).
- Secondly, in the same curved portions, when the curvature increases or decreases, the contour error profile may behave differently (S2 and S3, S5 and S6, S8 and S9).
- Lastly, the modification of control gains leads to modifying the resulting feedrate tracking. A small adjustment of the resulting feedrate can produce a large change in the contour error.

According to the above discussions, it can be concluded that the curvature characteristics and the control gain values are two main factors that are responsible for the presence and variations of contour error.

In addition, it would be significant to compare the contouring accuracy between the case (3) using the best fixed gain values K_a^{P*} and K_a^{F*}, and the case using the nominal gain values, $K_a^{P,n}$ and $K_a^{F,n}$, proposed by the machine tool manufacturer. In [Prévost 2011a], it can be found the nominal gain values in the position controller of the Mikron UCP 710 machine, as given in Table 3.3.

Table 3.3: Nominal gain values in position controller of Mikron UCP 710 machine

	Y	Z	A
$K_a^{P,n}$ (m/min/mm or rad/min/mrad)	1.5	3.5	1
$K_a^{F,n}$	1	1	1

Figs 3.8 shows that the obtained contour error of the case (3) using the best fixed gain is clearly much smaller than the one obtained by the case using the nominal gain values. In contrast, for the tool tracking error, the reverse is the case.

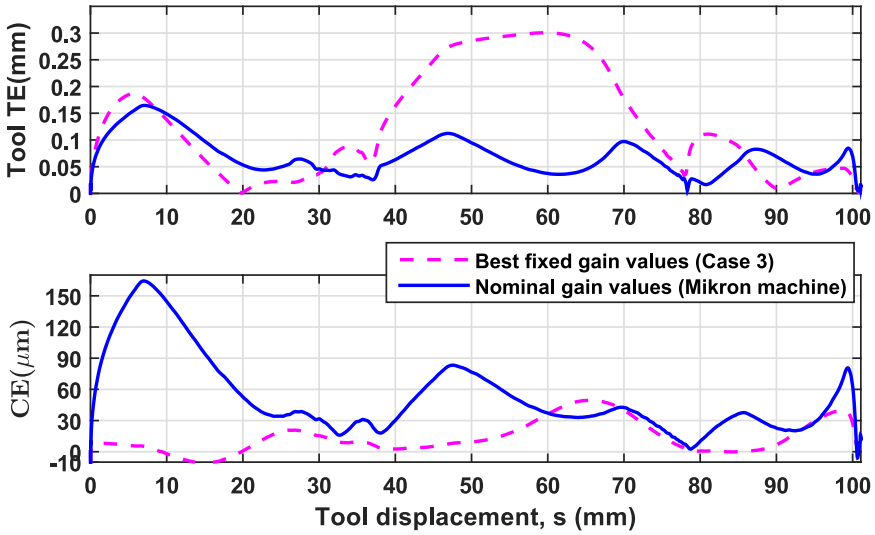

Figure 3.8: Comparing contour error profile of C3 and the nominal case

This can be explained that the nominal gain values are proposed by the machine tool manufacturer for mainly reducing the tool tracking error, expecting to indirectly reduce the contour error. However, the above comparison once again proves that the reduction of tool tracking error does not much reduce the contour error in machining free-form profile.

According to the above results and discussions, the response of the case (3), using the best fixed gain values K_a^{P*} and K_a^{F*}, is considered as the reference one, in order to compare with the performance of OGA.

1.4 Sensitivity of different control gains

The OGA is in fact applied to modify the values of K_a^P and K_a^F in the position controller, yielding the changes in the reference signal of the velocity loop and in the resulting feedrate profile after that, in order to compensate the contour error.

Thus, the two following questions come naturally: which control gain allows to achieve more effectivenesses with OGA? And the effectiveness occurs in which conditions? To answer these

questions, it is important to analyze the sensitivity of these control gains on the reference signal of the velocity loop.

The reference velocity signals from P and FFW actions, denoted as U^P and U^F respectively, are compared when one of the two control gains is kept fixed as its best fixed value, K_a^{P*} or K_a^{F*}, and the other is tuned with different values. For such a analysis, Fig. 3.9 compares U^P and U^F for Y axis.

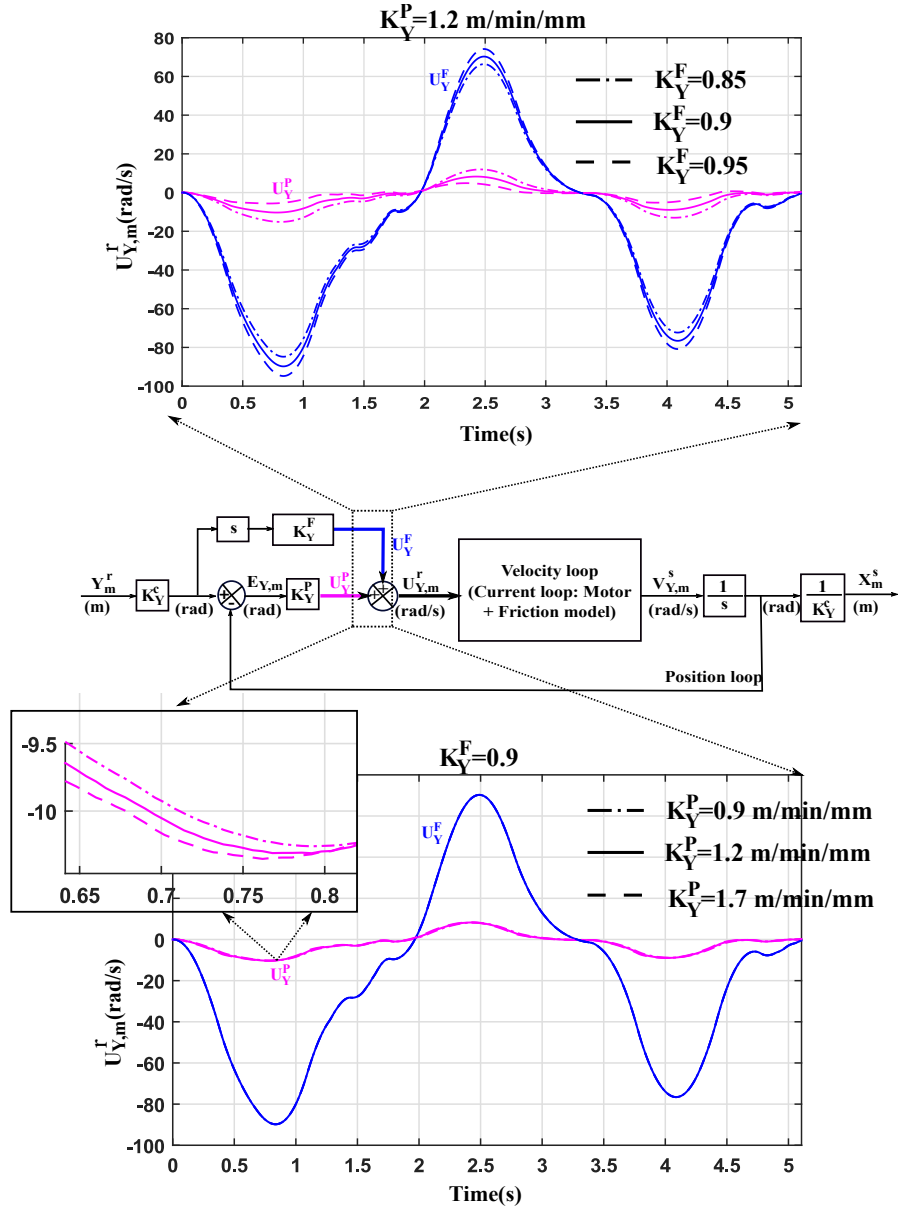

Figure 3.9: Comparing velocity reference of FFW and P actions, for Y axis

On top, Fig. 3.9 indicates that when K_Y^P is kept fixed, the modification of K_Y^F leads to clear modifications in both U^P and U^F. It can be seen that greater K_Y^F, greater U^F, smaller U^P; and vice-versa. The contribution of U^P to the reference velocity signal is much less than U^F.

At bottom, Fig. 3.9 shows that when K_Y^F is kept fixed, the modification of K_Y^P does not change U^F, but slightly modifies U^P. This is obvious because U^F only depends on the position setpoint and the value of K_Y^F. In these three cases of study, greater K_Y^P, greater U^P and vice-versa.

According to the above discussions, it can be said that K_a^F modification is more sensitive than K_a^P in adjusting the reference velocity signal. It can be predicted that K_a^F only needs to be slightly adjusted to achieve the benefit of OGA; while K_a^P needs to be much modified to have similar effect as the K_a^F adjustment. These predictions will be checked in § 3.

2 Configurations

To perform the cases of study of OGA(a) and OGA(b), it is needed to configure their parameters, as follows:

- The sampling period of the position loop of Mikron UCP 710 machine is $T_e = 6\ ms$.
- The axis kinematic limitations are taken from Table 2.2.
- The initial gain values (2.45) can be chosen as whatever value in the admissible gain range. In this work, it is proposed to start with the best fixed gains.
- The nominal motor current value I_a^n (2.37) is calculated based on the documentation of Siemens motor 1FT6084 - 8WF71 [Seimens 2005]. ΔI_a^{max} (2.38) is chosen as 2% of I_a^n ($k_I = 2\%$), in order to limit noise and large variations of the motor current.
- Other user-defined parameters, including the horizon length, the number of potential gain values, the length of the sinusoidal part in the pre-defined gain function of OGA(a), and the weighting factors, are discussed in detail below.

2.1 Horizon length

Chapter 2, Section § 3.3.1 proposed two ways to define the horizon length, either as a function of time or as a function of the trajectory geometry.

For OGA(a), the horizon length is defined considering the following two factors:

- Firstly, because the horizon in OGA(a) completely recedes without overlap, the optimal variable gains generated within a horizon are totally applied for controlling the trajectory over that horizon.
- Secondly, according to the three remarks in § 1.3, there is the fact that the free-form trajectory can be divided into many different portions, and each portion has its own curvature characteristics, e.g. either convex or concave or linear features. Moreover, one specified combination of the fixed control gains is suitable for controlling only one specified portion. Consequently, the horizon lengths should correspond to the above portions. Like this, the variable gains are optimized to be suitable with the curvature characteristics in those portions.

Due to the above two facts, it is proposed to define the horizon length of OGA(a) as a function of the trajectory geometry, which is illustrated in Fig. 3.10 and realized by Algorithm 1.

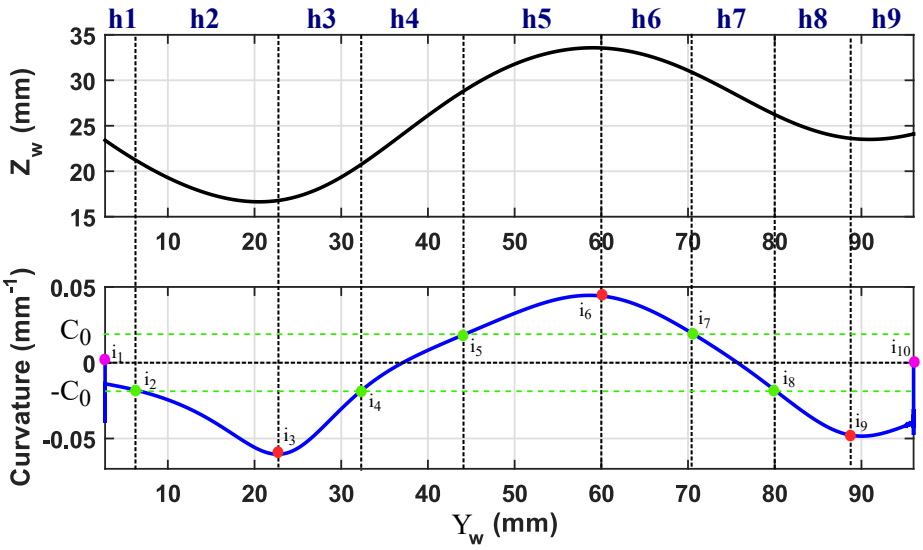

Figure 3.10: Illustration of the definition of geometry horizon for OGA(a)

Algorithm 1 Find the horizon length of OGA(a)

1: Calculate the curvature values of the reference trajectory, as illustrated in Fig. 3.10.
2: Choose the curvature value C_0 so that $[-C_0 : C_0]$ represents linear portions. In this work, $C_0 = 0.02\,(mm^{-1})$ is chosen.
3: Determine the total instants over the reference trajectory, denoted as n_i.
4: **for** $k = 1 : n_i$ **do**
5: Determine the first and last indexes on the trajectory, e.g. i_1 and i_{10} respectively.
6: Determine the indexes, e.g. $\{i_2, i_4, i_5, i_7, i_8\}$, that stay on the boundaries of the above linear portions.
7: Determine the critical indexes, e.g. $\{i_3, i_6, i_9\}$, that are the negative and positive curvature peaks.
8: **end for**
9: From the above obtained indexes, determine the horizon lengths as a function of tool displacement length as in (2.7).

For OGA(b), at each instant the objective is to find the optimal gain for the first future instant, while considering the future curvature characteristics of the trajectory.

If the horizon length of OGA(b) is chosen as adaptive as OGA(a), it may cause the wrong prediction effect. For example, in Fig. 3.11, the length of linear portion is much longer than that of the curved one. At the k instant, the tool stays at the linear portion. If OGA(b) takes the length of linear portion to evaluate the future trajectory containing mainly the curved portion, then the chosen gain, K_a^{k+1}, applied at the $k+1$ instant is not the optimal one. This induces the wrong effect of OGA(b).

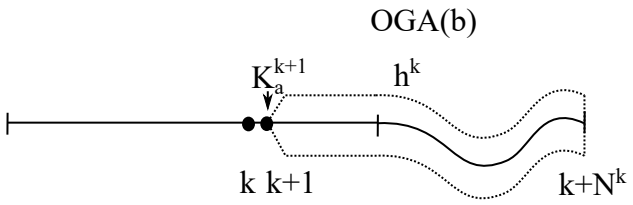

Figure 3.11: Illustration of the effect of the bad horizon length in OGA(b)

It is proposed to use a small fixed horizon length as a function of time for OGA(b), rather than the adaptive geometrical ones. With such a fixed temporal length, the number of instants

within each horizon is always the same, as discussed in Chapter 2, Section § 3.3.1. To firstly simulate OGA(b), the horizon length of OGA(b) is chosen constant and equal to 25 instants. This value means that if $T_e = 0.006$ s and the horizon has a constant feedrate of 3 m/min, then the geometrical length of such a horizon is 7.2 mm. The tuning effect of different horizon lengths in OGA(b) will be discussed in § 3.7.

2.2 Number of potential gain values

For both OGA(a) and OGA(b), the number of potential gain values is given by p_0 (2.53), that is in relation with the value of gain step in δ_a (2.56). For greater p_0, smaller δ_a, more possibilities are evaluated to obtain the optimal variable gains, but it is more time consuming. For example, if all of the five axes, $\{X, Y, Z, A, C\}$ and both K_a^P and K_a^F are involved into the optimization of OGA and if $p_0 = 20$, the number of tuning cases in each horizon will be $20^{10} = 1.024e^{13}$. This takes too huge computation time to finish the optimization of OGA. As a consequence, the OGA implementation would be intractable.

For the simulation of OGA(a) in § 3, p_0 will be chosen as 10. This value means that if all of the three axes $\{Y, Z, A\}$ and both of the two control gains are involved into the optimization, there are 10^6 tuning cases in each horizon, that is large enough to verify the effect of OGA(a). Because the horizon in OGA(a) is receded without overlap and the Algorithm 1 allows to generate 9 horizons over the reference trajectory, as seen in Fig. 3.10, thus there are 9.10^6 simulating iterations over the whole trajectory.

For OGA(b), the horizon is receded only one instant. Assume that the trajectory has N instants, there are $N-1$ horizons. In fact, the reference trajectory in Fig. 3.1.a has 852 instants, meaning that there are 851 iterations. Because the number of horizons in OGA(b) is much more than in OGA(a), p_0 in OGA(b) must be chosen less than in OGA(a), in order to have a reasonable computation time. Finally, $p_0 = 5$ will be chosen to simulate OGA(b) in § 3, that corresponds a total number of $851.5^6 \approx 13.10^6$ simulations.

2.3 Length of the sinusoidal part in the predefined gain function

For OGA(a) only, the length of the sinusoidal part, l_1 (2.46), allows to adjust the shape of gain profile, as seen in Fig. 2.16. Because the horizon length is adaptive, depending on the curvature characteristics of the free-form trajectory, as discussed in § 2.1, the length of the sinusoidal part should be adaptive also. However, how to choose its adaptive values is an open question. To firstly analyze the responses of OGA(a), the length of the sinusoidal part is chosen as 30% of the horizon length. According to (2.48), this means $l_1 = 0.3N^k$ ($k_l = 0.3$). The tuning effect of different values of l_1 will be discussed in § 3.6.

2.4 Weighting factor

The weighting factors, λ^P and λ^F (2.24), allow to manage the gain variation amount. Smaller λ^P and λ^K, greater gain variations $\Delta\mathbb{K}_a^P$ and $\Delta\mathbb{K}_a^F$ can be accepted respectively, and vice-versa.

In the OGA, the main objective is to obtain the contour error as small as possible, as long as its constraints are respected. If a large gain variation can satisfy this motivation, then it can be chosen. In the sequel, $\lambda^P = \lambda^F = 0$ is firstly chosen to simulate both OGA(a) and OGA(b). However, the effect of different weighting factor values on the gain variations and the resulting contour errors would also be worth discussion in § 3.8.

2.5 Summary of the configurations

To summarize, the configurations of both OGA(a) and OGA(b) are given in Table 3.4. Particularly, T_0 (2.55) will be a tuning parameter in the next section § 3.

Table 3.4: Configurations of OGA(a) and OGA(b)

Type	Parameters	Data source	Value	Unit
General	T_e	Mikron machine	0.006	s
	$[I_Y^n, I_Z^n, I_A^n]$ (2.37)	Motor documentation	[26.2, 26.2, 42.7]	A
	$[\Delta I_Y^{max}, \Delta I_Z^{max}, \Delta I_A^{max}]$(2.38)	User-defined	2% $[I_Y^n, I_Z^n, I_A^n]$	A
	λ^P, λ^F (2.24)	User-defined	0	-
	$K_a^{P,1}, K_a^{F,1}$(2.45)	User-defined	K_a^{P*}, K_a^{F*}	see Chap. 2,§ 3.2
	$K_a^{P,min,max}, K_a^{F,min,max}$	User-defined	see Table 2.2	see Table 2.2
OGA(a)	N^k, $\Delta s^{h,k}$(2.7)	Algorithm 1	(+) see below	instant, mm
	l_1(2.48)	User-defined	$0.3N^k$	instant
	p_0 (2.53)	User-defined	10	-
OGA(b)	N^k, $\Delta t^{h,k}$ (2.6)	User-defined	25, 0.144	instant, s
	p_0 (2.53)	User-defined	5	-

(+) For the desired trajectory in Fig. 3.1.a, there are 852 instants, corresponding to the total length of the tool displacement of $s^{end} = 101$ mm. The nine adaptive horizons generated by the Algorithm 1 have the adaptive N^k and $\Delta s^{h,k}$ as follows:

N^k(h1:h9)=[77, 56, 111, 116, 63, 84, 114, 87, 144] instant.

$\Delta s^{h,k}$(h1:h9) =[8.3126, 10.7076, 12.9685, 17.1182, 13.5307, 11.2934, 9.5413, 8.8023, 8.7577] mm.

3 Discussions

This section consists of analyzing not only the effects of the parameter configurations on the performance of OGA, but also the responses of OGA, including the obtained variable gain profile, the axis kinematic responses, the resulting feedrate and contour error profiles.

The cases of study of OGA(a), OGA(b) and the reference case are denoted and explained in Table 3.5.

Table 3.5: Cases of study for OGA with the classical control structure

Control strategy	Notation	Description
Classical control	r0	Using the best fixed gains
OGA(a)	OGA(a1)	Tuning K_a^P
	OGA(a2)	Tuning K_a^F
	OGA(a3)	Tuning both K_a^P and K_a^F
OGA(b)	OGA(b1)	Tuning K_a^P
	OGA(b2)	Tuning K_a^F
	OGA(b3)	Tuning both K_a^P and K_a^F

When only one control gain, K_a^P or K_a^F, is modified, the remaining control gain, K_a^F or K_a^P, is kept constant as its best fixed value, K_a^{F*} or K_a^{P*} in Table 3.2, respectively.

3.1 Impact of the variation factor

In the following, the effects of the variation factor T_0 (2.55) on the performance of both OGA(a) and OGA(b) are discussed. In fact, if a very high number of potential gains, p_0 (2.53), can be checked by OGA without suffering serious computation burden, T_0 has less meanings. However, as discussed in § 2.2, due to the inherent drawback of computation burden in the gridding solving techniques of OGA(a) and OGA(b), p_0 must be limited, e.g. as 10 and 5 for OGA(a) and OGA(b), respectively, to assure a reasonable computation time for the OGA optimization. Due to this fact, the value of T_0 is needed to regulate the gain step δ_a and the maximum gain variations. As a result, it allows to compensate inconveniences due to the limited number of potential gains. It can be argued that adjusting δ_a can also have the same effect as T_0. However, T_0 is preferred because it has more physical meanings than δ_a, when it directly shows the ratio between the maximum allowable gain variation and the admissible gain range.

The cases of study related to this parameter in OGA(a) and OGA(b) are illustrated in Fig. 3.12. In fact, T_0 is tuned over the range $[0.0001 : 1]$, however only five cases of study of OGA(a), including Ra1, Ra2, Ra3, Ra4 and Ra5, and five cases of study of OGA(b), including Rb1, Rb2, Rb3, Rb4 and Rb5, are shown, because they are enough to highlight the effect of the variation factor T_0 on the performance of OGA(a) and OGA(b).

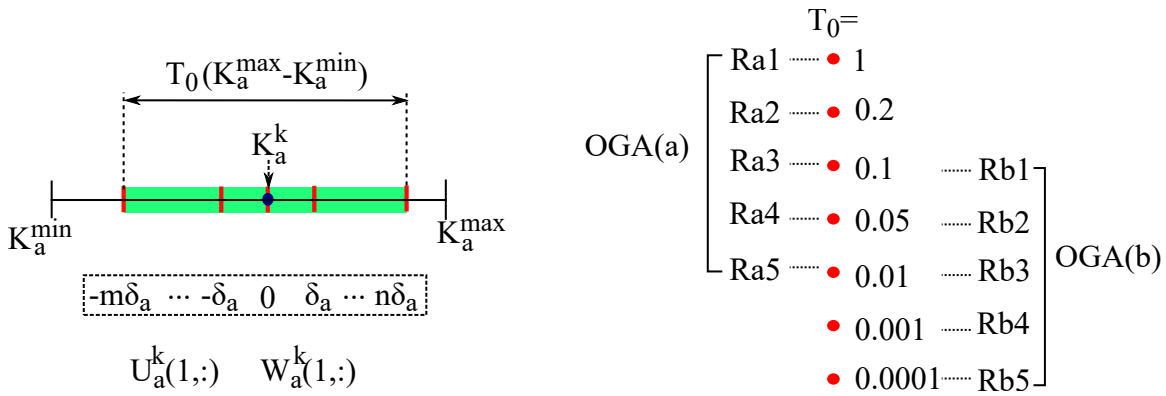

Figure 3.12: Illustration of the cases of study for the variation factor of OGA(a) and OGA(b)

With the admissible gain range in Table 3.1 and the number of potential gains in Table 3.4, the corresponding gain steps of the above cases of study of T_0 are calculated in (2.56) and given in Table 3.6 and Table 3.7. It can be seen that ranging from Ra1 to Ra5 and from Rb1 to Rb5, the variation factor T_0 is decreasing, consequently the gain steps, δ^P and δ^F, are decreasing also.

Table 3.6: Different gain steps in OGA(a)

	δ^P					δ^F				
	Ra1	Ra2	Ra3	Ra4	Ra5	Ra1	Ra2	Ra3	Ra4	Ra5
Y	0.0889	0.0178	0.00889	0.0044	0.000889	0.12	0.0240	0.012	0.0060	0.0012
Z	0.1111	0.0222	0.01111	0.0056	0.001111	0.2167	0.0433	0.02167	0.0108	0.00433
A	0.0222	0.0044	0.00222	0.0011	0.000222	0.1689	0.0338	0.01689	0.0084	0.00338

Table 3.7: Different gain steps in OGA(b)

	δ^P					δ^F				
	Rb1	Rb2	Rb3	Rb4	Rb5	Rb1	Rb2	Rb3	Rb4	Rb5
Y	0.04	0.02	0.004	0.0004	0.00004	0.054	0.027	0.0054	0.00054	0.000054
Z	0.05	0.025	0.005	0.0005	0.00005	0.098	0.0488	0.0098	0.00098	0.000098
A	0.01	0.005	0.001	0.0001	0.00001	0.076	0.038	0.0076	0.00076	0.000076

According to the configurations in § 2, the OGA(a) and OGA(b) are performed with the above different cases of study of T_0. The obtained mean contour errors, $\bar{\varepsilon}$ (3.4), are given in Table 3.8.

Table 3.8: The mean contour errors of OGA(a) and OGA(b)

		$\bar{\varepsilon}(\mu m)$				
r0		14.4				
OGA(a)		Ra1	Ra2	Ra3	Ra4	Ra5
	OGA(a1)	**4.9**	9.4	10.6	12.4	12.8
	OGA(a2)	12.2	2.8	**1.4**	3.5	4.2
	OGA(a3)	4.2	1.2	**0.7**	2.3	3.5
OGA(b)		Rb1	Rb2	Rb3	Rb4	Rb5
	OGA(b1)	17.1	**4.2**	4.8	12.3	12.5
	OGA(b2)	18.6	10.4	6.3	**0.68**	3.8
	OGA(b3)	10.2	2.1	**0.23**	0.4	2.5

There are important remarks based on the results in Table 3.8, as follows:

- Almost all of the cases of study with OGA(a) and OGA(b) produce smaller contour error than the reference case r0, except for OGA(b1) and OGA(b2) in Rb1. Based on these quantitative evaluations, it is hard to explain such low performances. However, when the obtained gain profiles and jerk responses are analyzed in the section § 3.4, these worse results of OGA will be clarified.

- OGA(a3) and OGA(b3) are the best cases of OGA(a) and OGA(b), respectively. It can be explained that the OGA(a3) and OGA(b3) have more possibilities in tuning the gains, due to the combination of all potential gains of both K^P and K^F.

- In OGA(a), OGA(a1) has smaller contour error than OGA(a2) with Ra1; while in all of the remaining cases the reverse is the case. In OGA(b), OGA(b1) is better than OGA(b2) in the cases of Rb1, Rb2 and Rb3; while in Rb4 and Rb5, OGA(b2) is better than OGA(b1).

  The above phenomenons can be explained by the sensitivity of the K^P and K^F values over the reference velocity signal, that has been discussed in § 1.4, i.e. Fig. 3.9. It can be said that in order to see the effects of the K^P adjustment, the gain variations of K^P should be large enough to be able to regulate the reference velocity signal and of course the feedrate, in order to further compensate the contour error. In the opposite, the small

gain variations of K^F are suitable to adjust the axis velocity and the feedrate in a delicate manner to compensate the contour error.

- If the gain variations are too small, as in Ra5 and Rb5, they generate too small effects on the axis velocity and the feedrate. As a consequence, the contour error is not much compensated in these cases.

- For OGA(a), the best case of OGA(a1) is Ra1, while the best cases of OGA(a2) and OGA(a3) are Ra3. For OGA(b), the best cases of OGA(b1), OGA(b2) and OGA(b3) are namely Rb2, Rb4 and Rb3. Based on these facts, it is hard to find out a clear method to choose T_0, in order to achieve the best performances of OGA(a) and OGA(b).

  According to the above results and for the given axis drives, it can only be pointed out two following facts:

  In OGA(a), for adjusting K^P, $T_0 = 1$ in Ra1 should be chosen. It means that the maximum gain variation of K^P should be equal to the admissible gain range. For adjusting K^F, its maximum gain variation should be 10% of its admissible gain range, as in Ra3.

  In OGA(b), the maximum gain variations of K^P and K^F should be chosen as within 5% and 0.1% of their admissible gain range, respectively.

Because OGA(a) and OGA(b) have different ways and characteristics in generating the optimal gain profile, it is hard to compare them within the same condition. It is proposed to compare between their best cases in Table 3.8 (in bold). The improvement and the computation time of these best cases are given in Table 3.9. There are some important remarks as follows.

- In general, the best performance of OGA(b) is better than that of OGA(a). However, the computation time of the former is much longer than the latter.

- For OGA(a), the best cases of OGA(a1), OGA(a2) and OGA(a3) improve the contouring accuracy with a contour error reduction of around 67%, 90% and 95%, respectively, compared with the reference case r0.

  With the same computation time, OGA(a2) is enhanced around 71% as compared with OGA(a1). Although having 50% of improvement compared with OGA(a2), OGA(a3) takes much more computation time than OGA(a2), 1.5 days compared with 12.5 minutes.

Table 3.9: The best performances of OGA(a) and OGA(b)

		$\bar{\varepsilon}(\mu m)$	Improvement	Computation time
r0		14.4	-	-
OGA(a)	OGA(a1)	4.9	65.97% r0	$\simeq$ 12.5 min
	OGA(a2)	1.4	90.28% r0	$\simeq$ 12.5 min
			71.43%a1	
	OGA(a3)	0.7	95.14% r0	$\simeq$ 2260 min ($\simeq$ 1.5 days)
			50%a2	
OGA(b)	OGA(b1)	4.2	70.83% r0	$\simeq$ 6.5 h
	OGA(b2)	0.68	95.28% r0	$\simeq$ 6.5 h
			83.81%b1	
	OGA(b3)	0.23	98.40% r0	$\simeq$ 45.5 h ($\simeq$ 1.89 days)
			66.18%b2	
			67.14%a3	

- It can be seen that the best cases of OGA(b1), OGA(b2) and OGA(b3) can respectively reduce around 70%, 95% and 98% the contour error of the reference case r0. The OGA(b2) takes the same computation time as the OGA(b1), but the former reduces around 83% the contour error of the latter. Meanwhile, the contouring accuracy of OGA(b3) is improved around 66% compared with OGA(b2), but the former takes 7 times longer to finish the computation than the latter.

- The best case of OGA(b), OGA(b3) in Rb3, has improved 67.14% the contouring accuracy of the best case of OGA(a), OGA(a3) in Ra3.

From the above results and discussions, it can be said that to have a reasonable computation time and a good contouring accuracy, it should be chosen OGA(a2) or OGA(b2). However, if in off-line execution the computation time is normally not limited, and the contouring accuracy has the highest priority, it is clear that OGA(a3) and OGA(b3) are preferred.

3.2 Impact of variable gains on contour error profile

This section is to illustrate the variable gains and contour error profiles, obtained from the best performances of the cases of study of T_0 in § 3.1, represented in Table 3.9. Theses responses are plotted in function of the trajectory in the workpiece space.

Fig. 3.13 gives a general view for the best variable gains and contour error profiles of OGA(a1) in Ra1, OGA(a2) and OGA(a3) in Ra3.

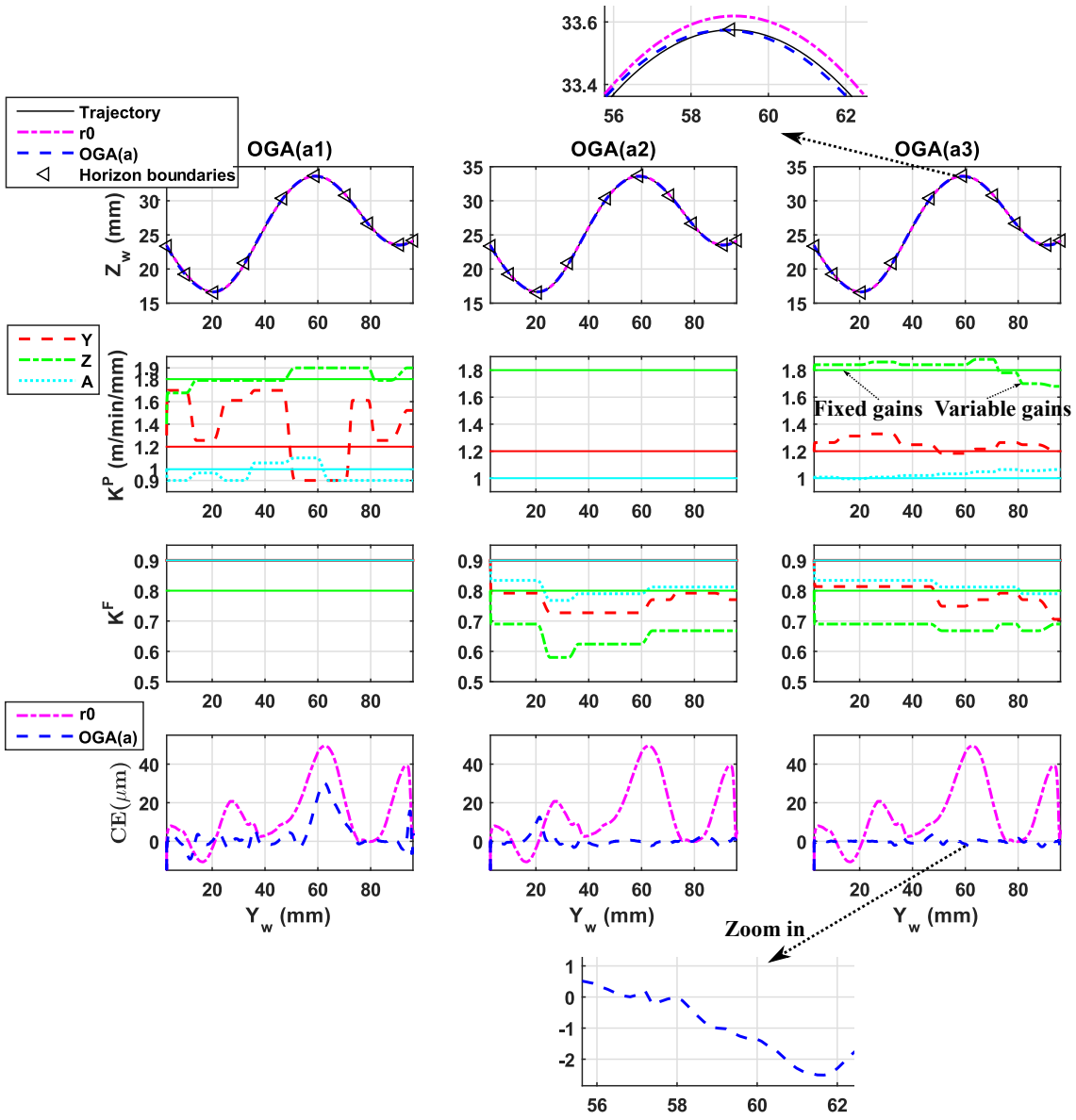

Figure 3.13: Resulting variable gains and contour error profiles of OGA(a)

For OGA(a), the horizon boundaries are also plotted to easily check the variable gain formulation due to the predefined gain function within each horizon.

Fig. 3.14 does the same but for the best cases of OGA(b), meaning OGA(b1) in Rb1, OGA(b2) in Rb4 and OGA(b3) in Rb3.

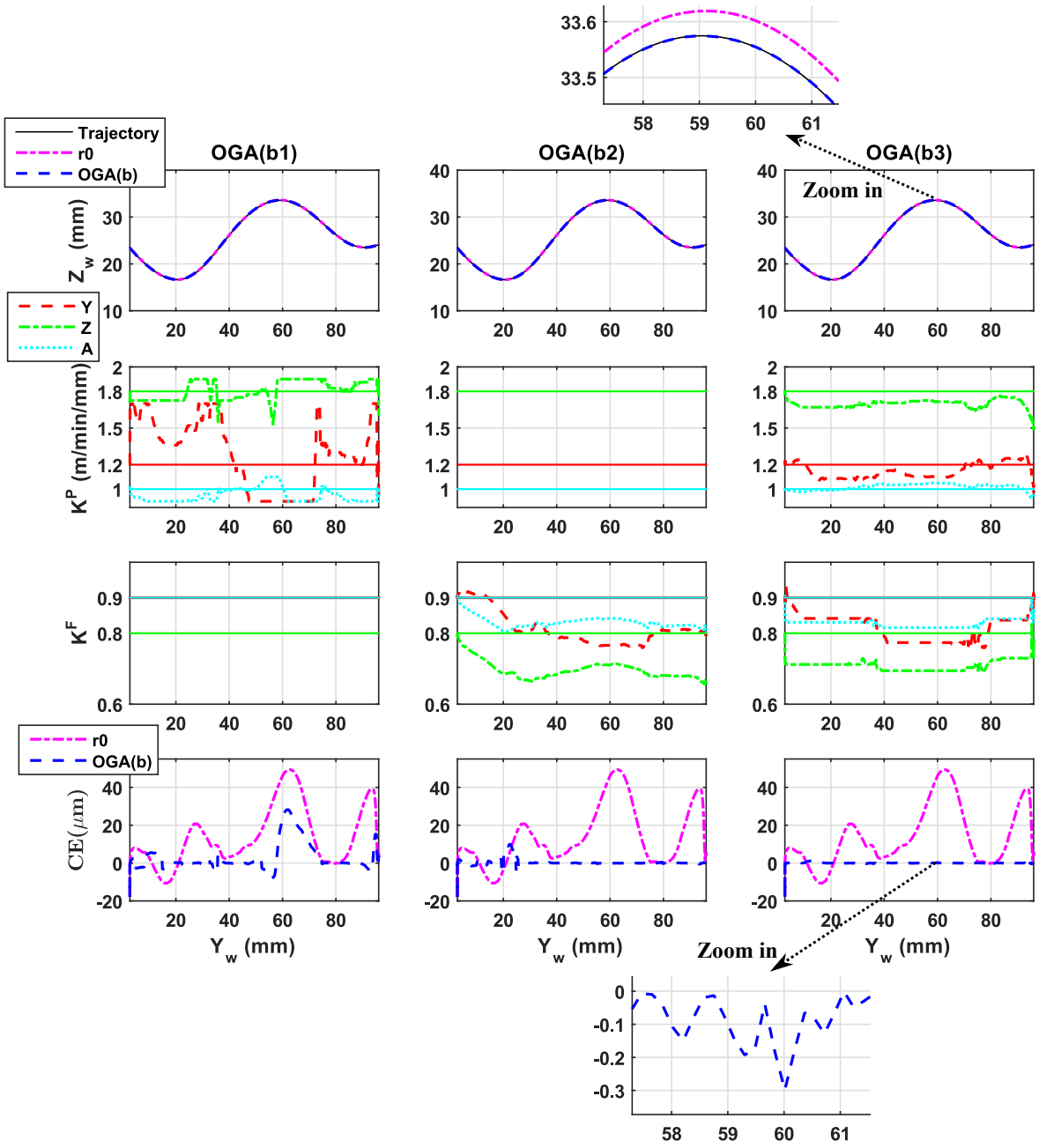

Figure 3.14: Resulting variable gains and contour error profiles of OGA(b)

It can be seen that all of the variable gains respect the admissible gain range in Table 3.1. The gain variations of OGA(a3) and OGA(b3) are slightly less than the remaining cases. It can be understood that, when only one control gain is adjusted, it must be much changed in order to compensate the contour error. While, if both K^P and K^F are adjusted, a small modification of each gain is enough to have the effect of contour error compensation.

These variable gains affect the contour error profiles, as seen in the three plots at the bottom of Fig. 3.13 and Fig. 3.14. From OGA(a1) to OGA(a3) and from OGA(b1) to OGA(b3) the contour error profile is improving in a way that it is approaching the zero line over the whole trajectory. In OGA(a1) and OGA(b1), there are contour error peaks at the region around $Y_w = 60\ mm$. While, in OGA(a2) and OGA(b2) there are smaller contour error peaks at the region around $Y_w = 20\ mm$. These contour error peaks are all smaller than the ones in the reference case. In OGA(a3) and OGA(b3), all of the contour error peaks appeared in the previous cases have been removed. The zoomed views indicate that the contour error of OGA(a3) and OGA(b3) only fluctuates several micrometers and several tenth of micrometers, respectively, around the zero line of contour error.

It can generally be said that the optimal variable gains of either OGA(a2) or OGA(a3) or OGA(b2) or OGA(b3) have mostly overcome the challenges due to the high curvature along the reference trajectory, in order to generate a small contour error over the whole free-from trajectory.

3.3 Impact of variable gains on axis kinematic and motor current responses

In this section, OGA(a3) in Ra3 and OGA(b3) in Rb3, presented in § 3.1, are employed to highlight the effect of the obtained variable gains on the axis kinematic and current responses.

Fig. 3.15 and Fig. 3.16 show that all of the axis kinematic responses respect their limitations in Table 2.2. With a general view, when the control gains are adjusted, there are small variations of the axis position, velocity and acceleration, but a large variation of the axis jerk, especially in OGA(b3).

A comparison of the axis kinematic responses between OGA(a3) and OGA(b3), e.g. Y axis, is

illustrated in Fig. 3.17. It can be noticed that the gain profile of OGA(b3) fluctuates more than the one of OGA(a3). As a consequence, the axis kinematic responses of OGA(b3) is much more excited than that of OGA(a3), especially the axis jerk. While the axis position, velocity and acceleration responses are quite the same as the reference ones, the response of axis jerk is more excited.

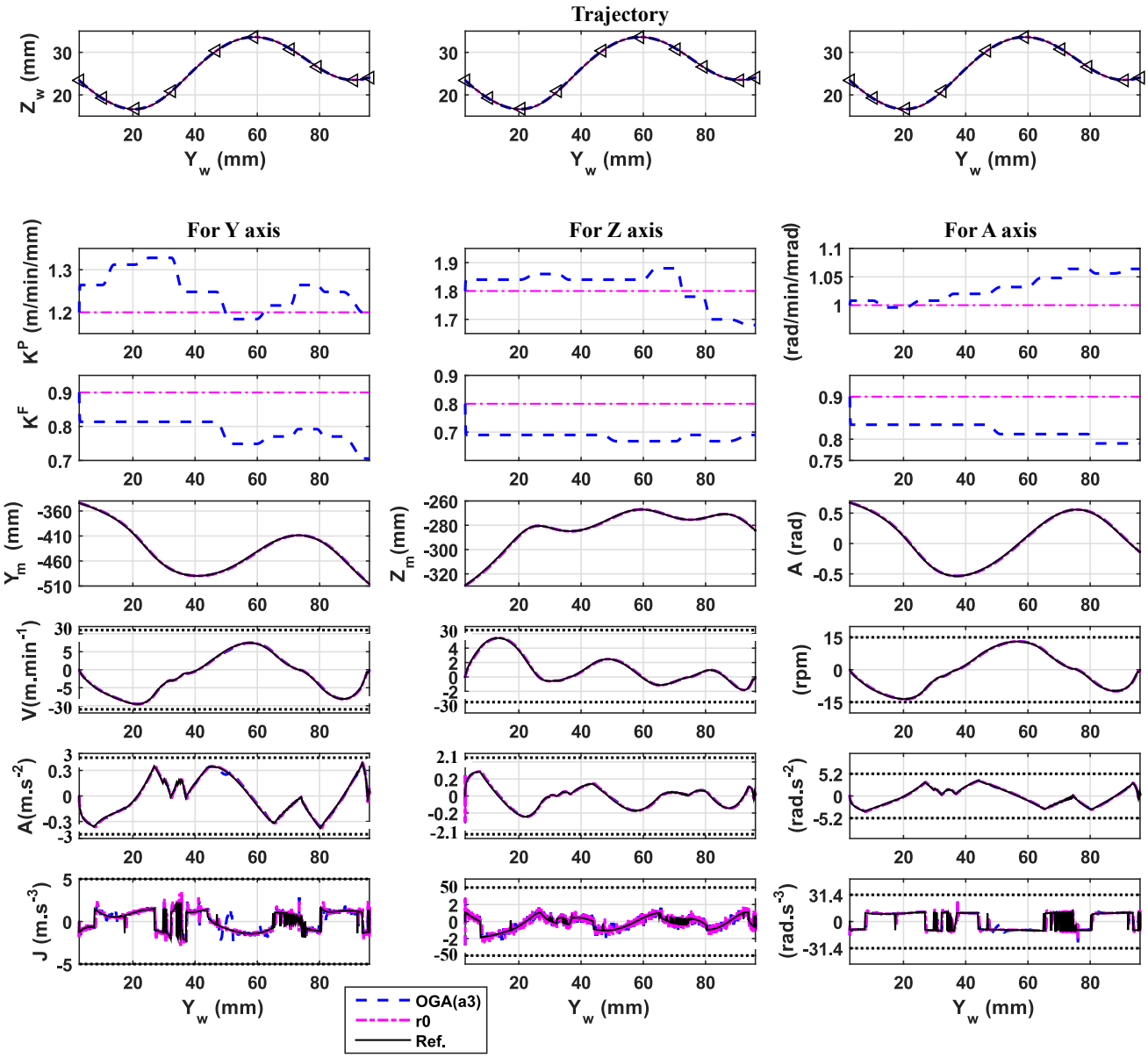

Figure 3.15: Axis kinematic responses of OGA(a3)

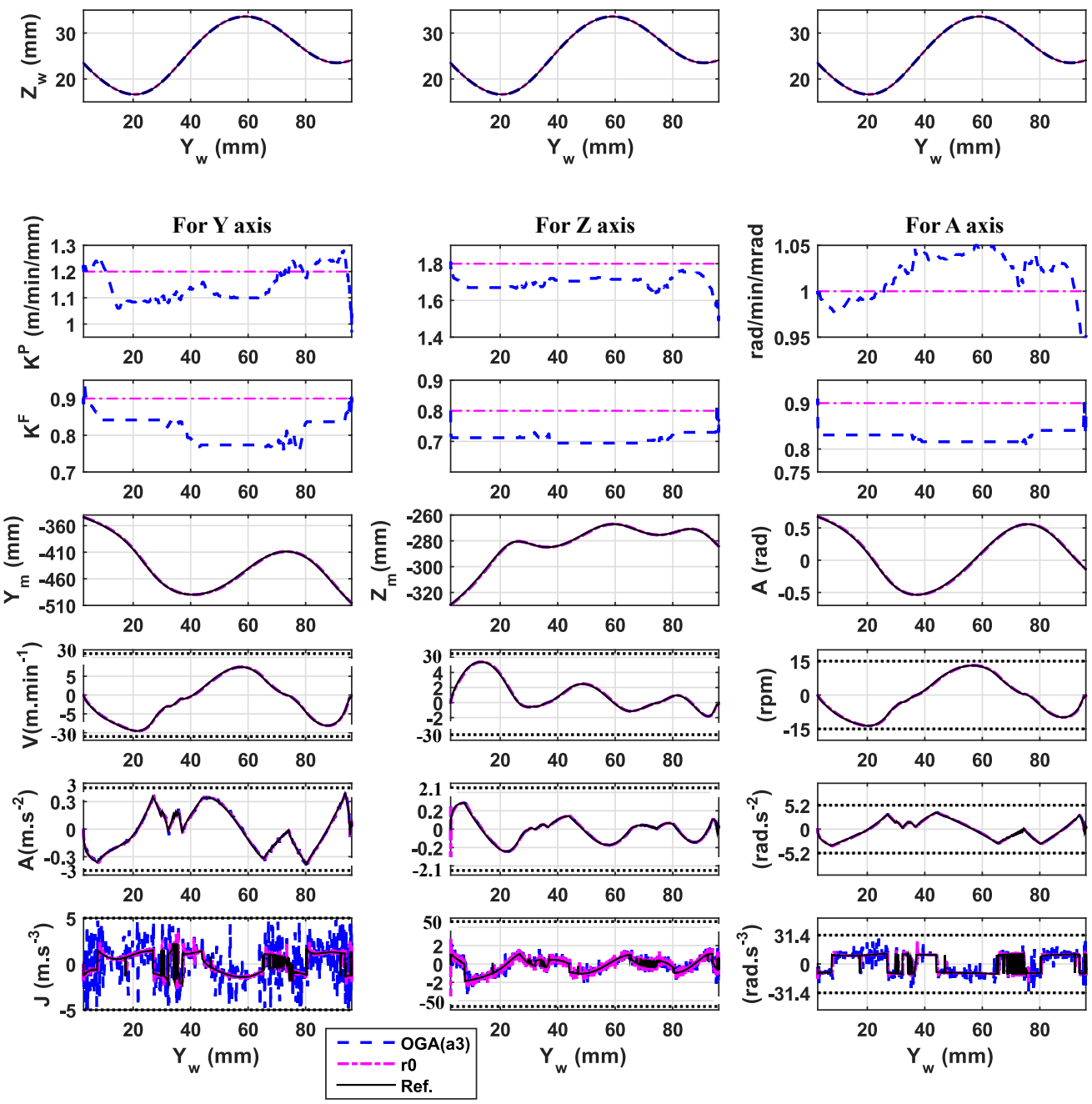

Figure 3.16: Axis kinematic responses of OGA(b3)

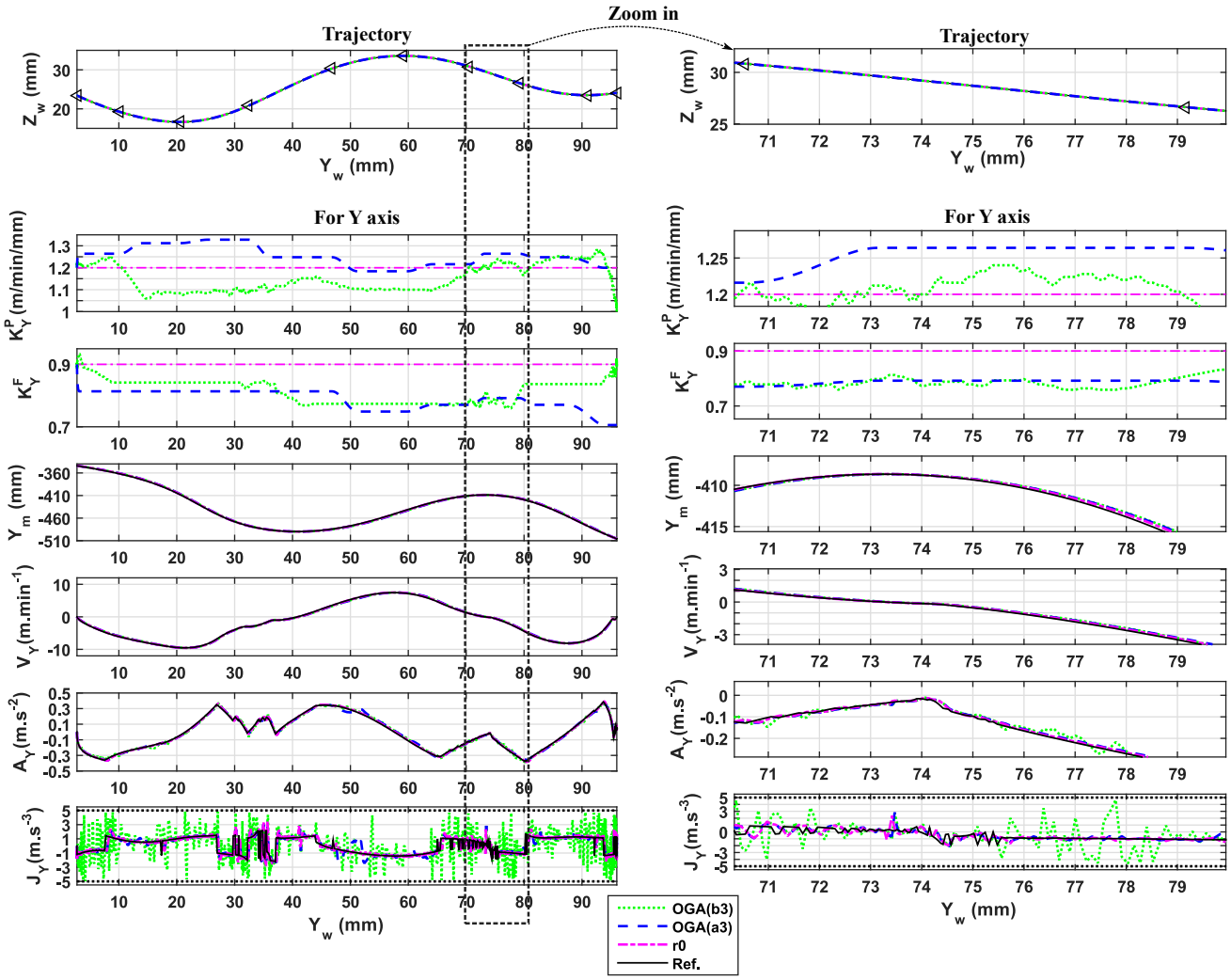

Figure 3.17: Comparison of the variable gains and kinematic responses of OGA(a3) and OGA(b3): Y axis case

Moreover, the motor current responses of the reference case r0, OGA(a3) and OGA(b3) are compared in Fig. 3.18. Their constraints in (2.33) are verified. The motor current responses of OGA(a3) and OGA(b3) are very similar to that of the reference case. The current increments are less than 2% of the nominal current values, which have been set in the initial configuration, as shown in Table 3.4.

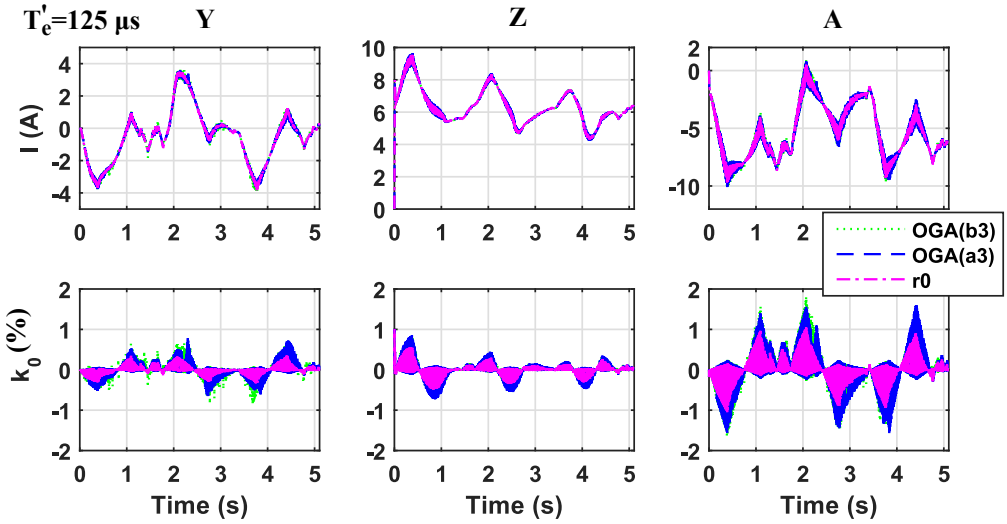

Figure 3.18: Motor current responses of r0, OGA(a3) and OGA(b3)

3.4 Relation between variable gains, jerk response and contour error

This section consists in answering the question: why OGA(a1) in Ra1 ($T_0 = 1$) is better than in Ra2 ($T_0 = 0.2$), while OGA(b1) in Rb1 ($T_0 = 0.1$) is worse than in Rb2 ($T_0 = 0.05$), in which OGA(b1) in Rb1 is even worse than r0?

Fig. 3.19 shows the obtained variable gains, axis jerk responses (e.g. Y axis) and the resulting contour error profiles of OGA(a1) in Ra1 and Ra2. There are two important remarks as follows.

Firstly, large gain variation of K_a^P in OGA(a1) does not much excite the jerk response. As it can be seen, the gain variations in Ra1 are much larger than in Ra2. This is because T_0 in Ra1 is five times more than in Ra2. This leads to the fact that the jerk variations in Ra1 are more significant than in Ra2. However, the axis jerk variations in Ra1 are still much less than its limitation and they only occur at the sinusoidal gain part in each horizon.

Secondly, due to the sensitivity contribution of K_a^P in the position controller discussed in § 1.4, it can be stated that it would be good to have a large variation of K_a^P to adjust the reference velocity signal of the velocity loop and the feedrate after that, in order to compensate the contour error.

Clearly, the above two facts prove that for the contour error compensation in OGA(a1) the large gain variation of K_a^P in Ra1 is better than the small one in Ra2.

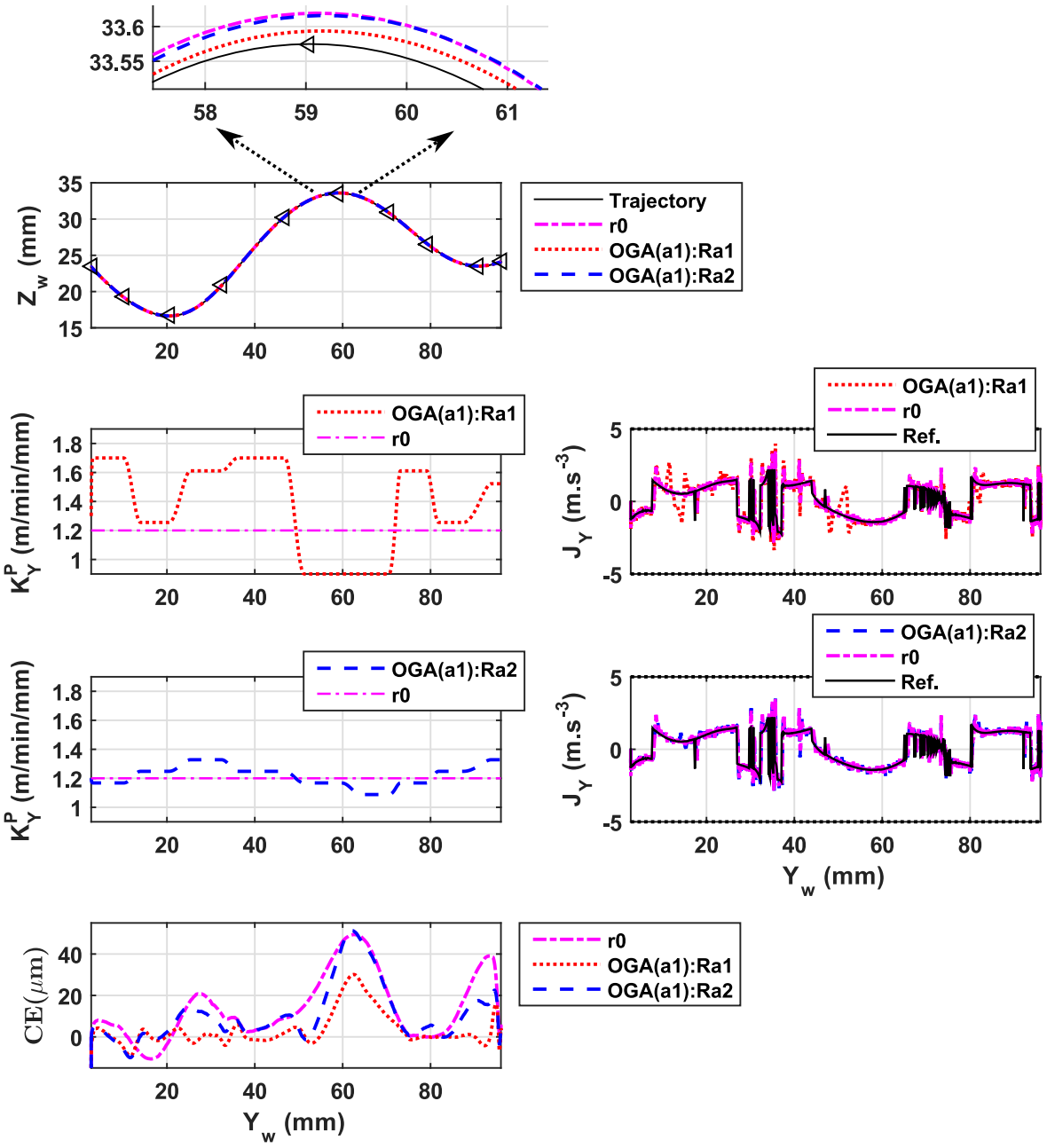

Figure 3.19: Analysis of OGA(a1) in Ra1 and Ra2

Fig. 3.20 is similar to Fig. 3.19, but for OGA(b1) in Rb1 and Rb2. As it can be seen, a small gain variation of OGA(b1) in Rb2 can much excite the axis jerk responses, approaching the limitations. Thus, the large gain variation of OGA(b1) in Rb1 has obviously higher possibility to violate the jerk limitation.

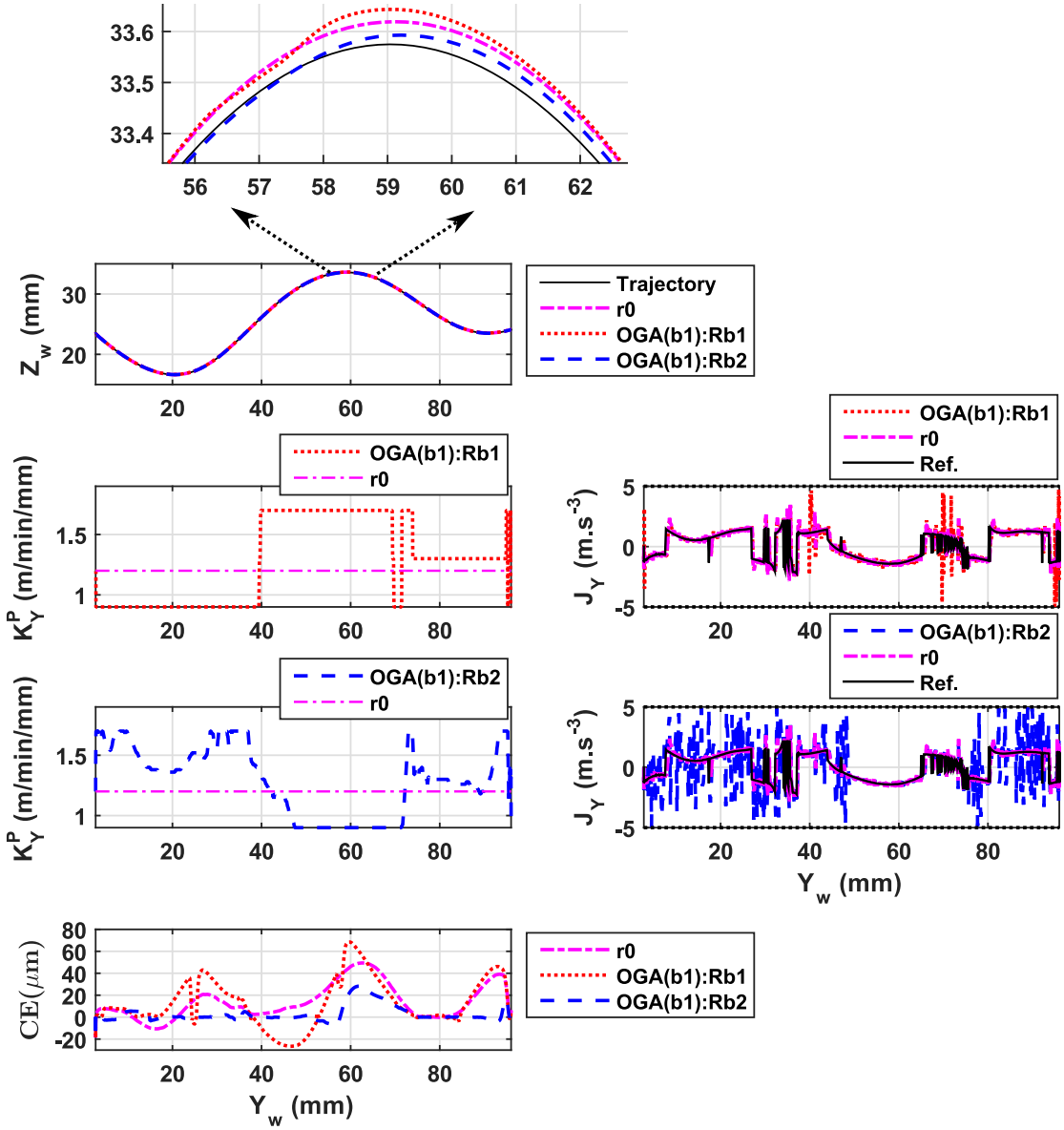

Figure 3.20: Analysis of OGA(b1) in Rb1 and Rb2

Consequently, if a large gain variation for the minimum contour error does not satisfy the constraints, especially the axis jerk limitation, the OGA will remove this optimal choice. Therefore, the obtained gain profile of OGA(b1) in Rb1, containing many constant gain portions and several large gain jumps, is not the optimal one for the contouring following anymore and even worse than the classical case. Meanwhile, the small gain variations in Rb2 can both compensate the contour error and still respect the jerk limitation. These facts explain why the contour error compensation of OGA(b1) in Ra1 is worse than in Ra2 and even worse than r0.

3.5 Relation between variable gains, feedrate, tool tracking error and contour error

This section is dedicated to analyze the relation between the variable gains, the feedrate, the tool tracking error and the contour error in the workpiece space. Fig. 3.21 and Fig. 3.22 respectively illustrate the concerned responses of OGA(a3) in Ra3 and OGA(b3) in Rb3, as represented in § 3.1 As it can be observed, the adjustment of the control gains generally changes the tool positioning behavior, including the feedrate, tool tracking error and contour error. Moreover, a small change of the feedrate can also result in a significant contour error reduction.

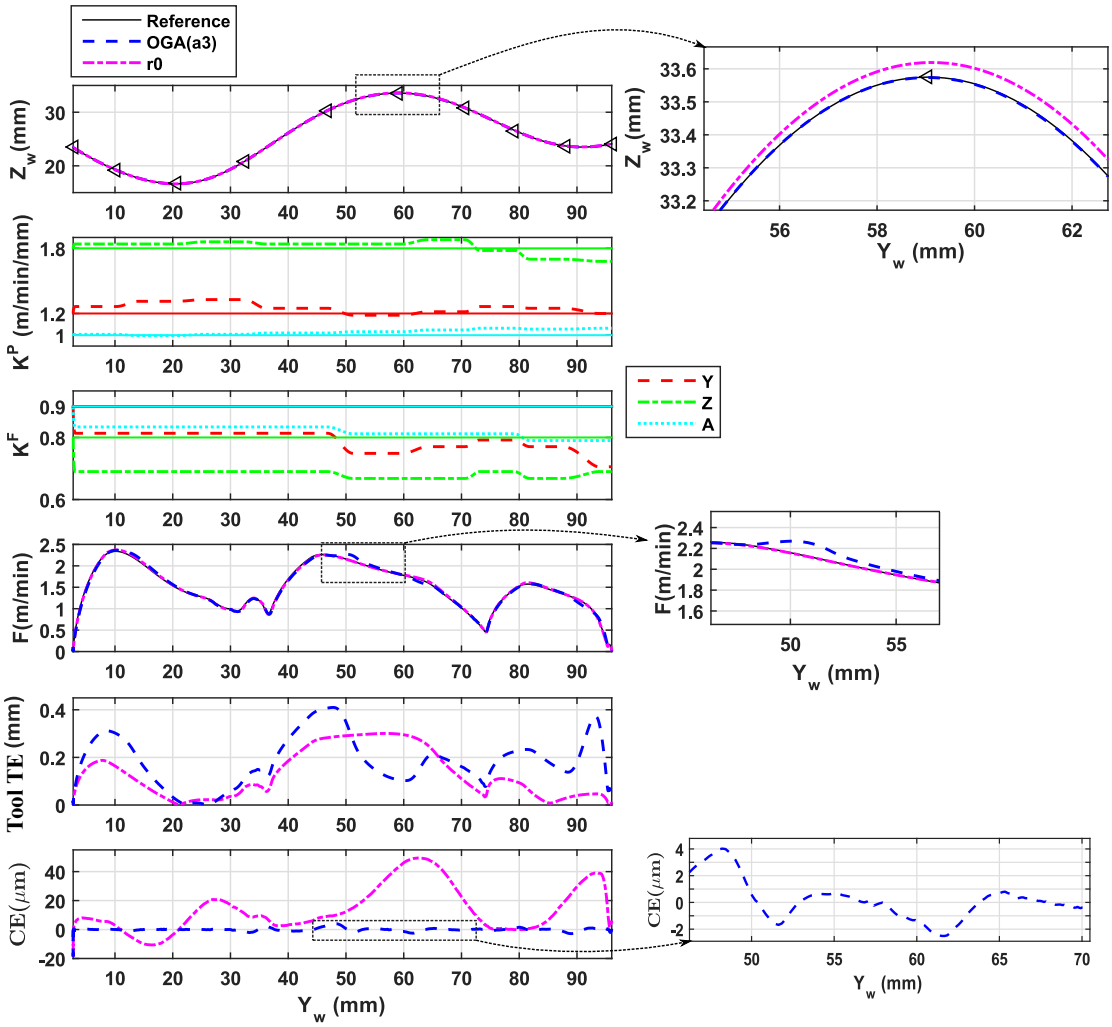

Figure 3.21: Impact of variable gains of OGA(a3) on the resulting feedrate, tool tracking error and contour error

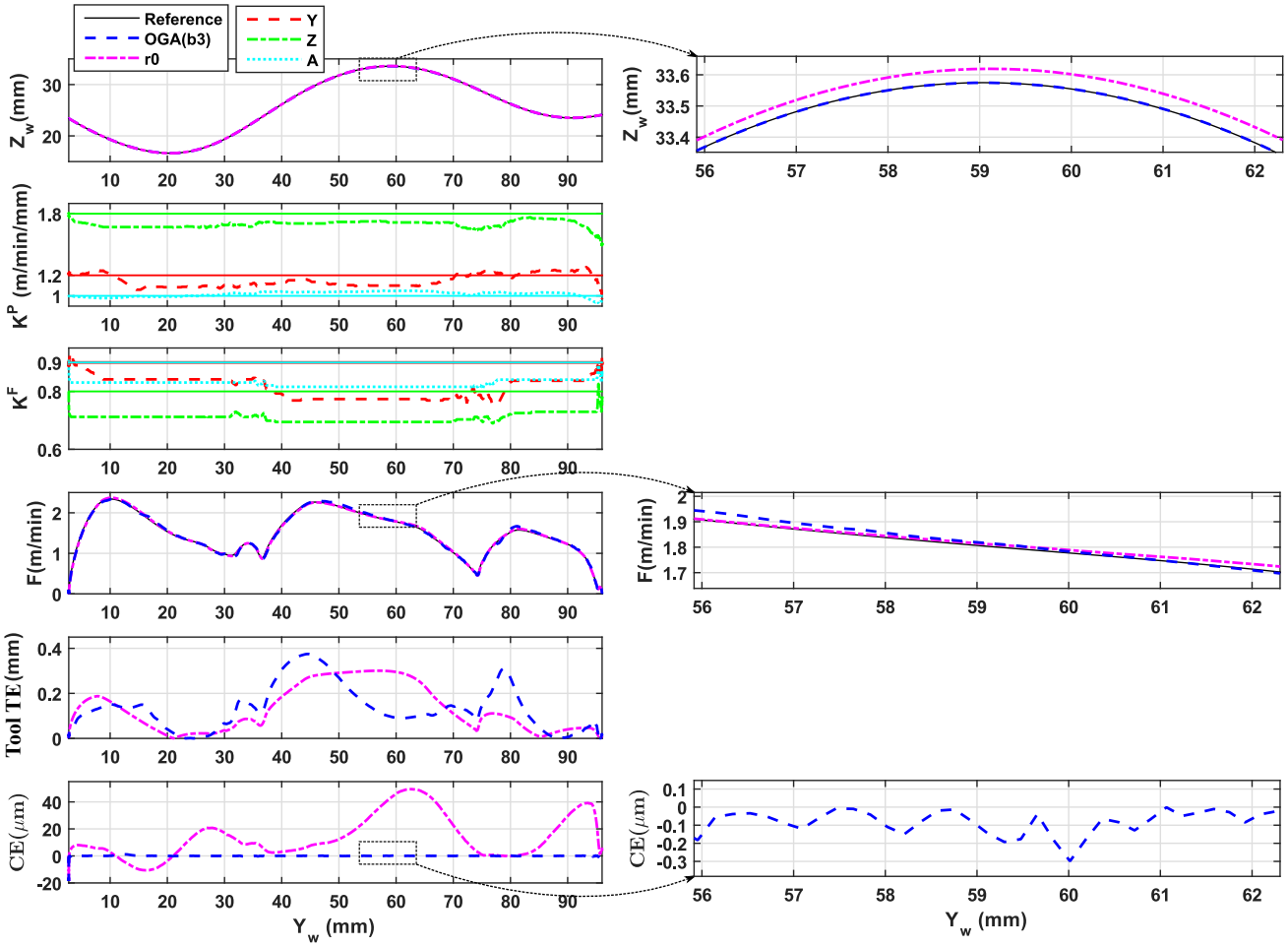

Figure 3.22: Impact of variable gains of OGA(b3) on the resulting feedrate, tool tracking error and contour error

While the contour error is much reduced by OGA, the resulting tool tracking error of OGA is sometimes increased as compared with r0. This confirms the fact that a zero contour error can be obtained even if the tool tracking error still exists.

The zoomed views in Fig. 3.21 and Fig. 3.22 prove that both OGA(a) and OGA(b) make the tool displacement very accurate over the peak of the convex curve zone, which is one of the critical regions in machining the free-form trajectory.

The comparison of the contouring accuracy between OGA(a3) and OGA(b3) is illustrated in detail through Fig. 3.23. Over the trajectory, the contour error of OGA(b3) mostly fluctuates within the limit of 1 μm, while the OGA(a3) has some contour error peaks from 2 to 4 μm.

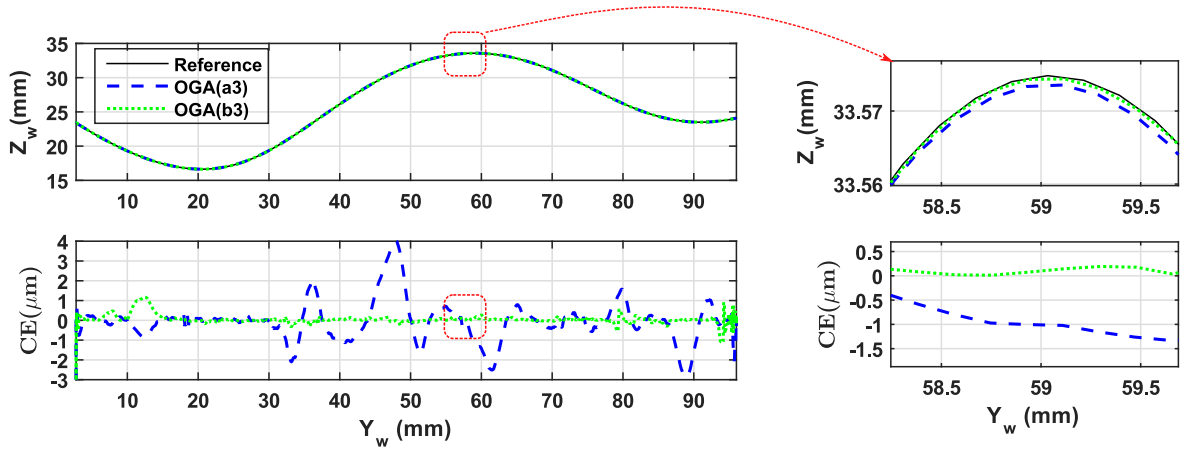

Figure 3.23: Comparison of the contouring accuracy between OGA(a3) and OGA(b3)

In addition, Fig. 3.24 shows another ways to evaluate the contouring accuracy of the above two cases.

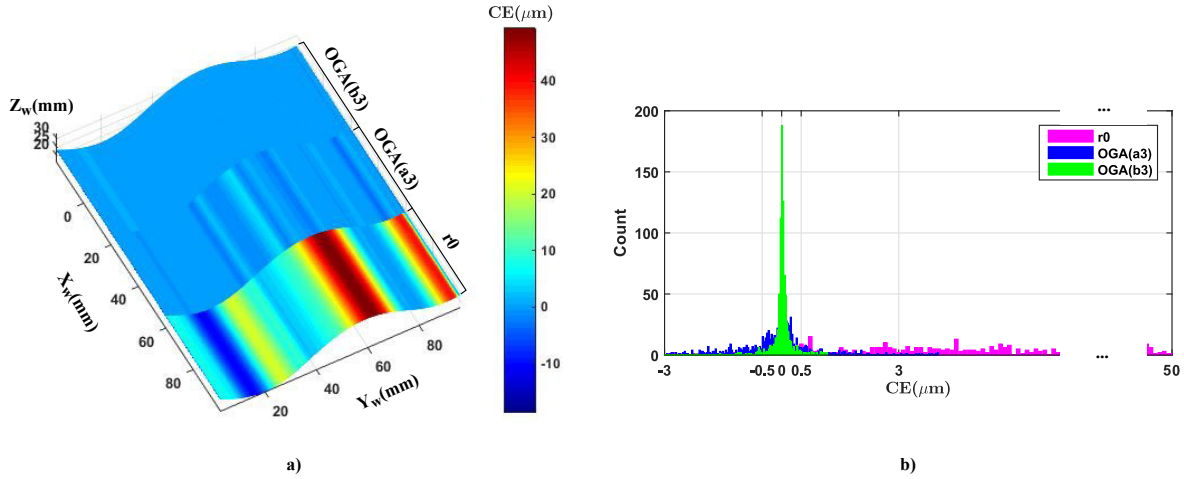

Figure 3.24: a) Illustration of the contour error along the machining surface; b) Histogram of the contour error

Fig. 3.24.a shows that ranging from r0, OGA(a3) to OGA(b3), the contour error over the simulated machining surface has been remarkably reduced.

Fig. 3.24.b illustrates that the contour error of the reference case r0 mainly ranges from 0.5 to 50 μm. While, OGA(a3) and OGA(b3) have a centered normal distribution of contour error, ranging from -3 to 3 μm and from -0.5 to 0.5 μm, respectively.

3.6 Impact of the sinusoidal part of OGA(a)

The effects in tuning different lengths l_1 (2.48) of the sinusoidal part in the predefined gain function (2.46) of OGA(a), are now discussed. Note that l_1 is kept the same over all of the adaptive horizons of OGA(a).

Three cases of study related to the length l_1 of OGA(a2) in Ra3 from § 3.1 are given in Table 3.10. Remember that OGA(a2) in Ra3 of Table 3.8 has used $l_1 = 0.3N^k$.

Table 3.10: Cases of study for the sinusoidal part

Control strategy	Notation	l_1	$\bar{\varepsilon}(\mu m)$	Improvement
Classical control	r0	-	14.4	-
OGA(a2)	L1	$0.1N^k$	5.7	60.42%r0
	L2	$0.3N^k$	1.4	90.28%r0 75.44%L1
	L3	$0.8N^k$	1.4	90.28%r0

As it can be seen, the above three cases of study have compensated the contour error much more than the reference case r0. L1 reduces about 60% the contour error of r0. L2 and L3 similarly reduce about 90% and 75 % the contour error of r0 and L1, respectively.

In addition, the effects of different lengths of the sinusoidal part on the obtained variable gain profile, the axis jerk responses and the contour error profile are illustrated in Fig. 3.25.

As shown in Fig. 2.16, longer l_1, smoother gain profile, and vice-versa. Because L1 has the smallest l_1 in the three cases, its gain profile contains sharp changes. As it can be noticed, several sharp gain variations in L1, even with small amplitudes, can also lead to the corresponding large peaks in the axis jerk responses. This implies that other large and sharp changes of the optimal gains in L1 have been eliminated from the OGA optimization due to the constraints of axis jerk. Therefore, the gain profile of L1 is less flexible and has less optimal variable gains, leading to the fact that its contour error is not much compensated.

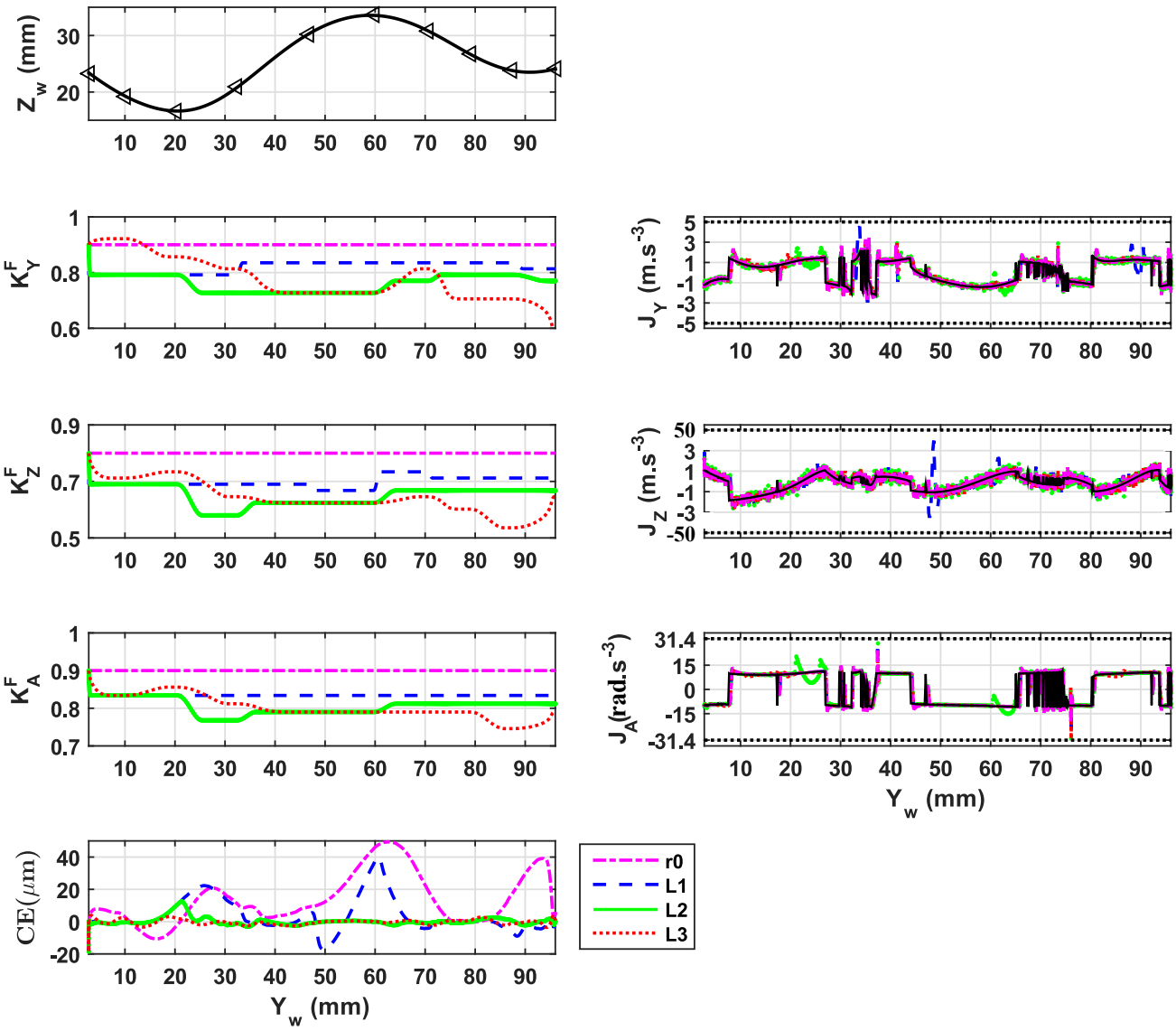

Figure 3.25: Impact of the sinusoidal part of gain function on OGA(a2)

In contrast, L2 and especially L3 have smoother gain profile and less jerk variations than L1. This means that the last two cases can generate more optimal gains than the first case. As a result, the contour error in L2 and L3 is reduced much more than in L1. Moreover, a smaller contour error peak of L2 around $Y_w = 20\ mm$ has been removed in L3.

According to the above discussions, it can be concluded that longer length of the sinusoidal part, more flexible profile of the variable gains, less excitations of the axis jerk responses, and more contour error reductions. However, how to globally choose the optimal length of the sinusoidal

part for the different adaptive horizons is still an open question. Up to now, it is still needed to do some tests and choose the best fixed length of the sinusoidal part for all of the adaptive horizons of OGA(a).

3.7 Impact of the horizon length of OGA(b)

This section refers to the tuning effect of different values of the fixed horizon length on the performance of OGA(b).

Four cases of study related to the horizon length of OGA(b1) in Rb2 from § 3.1 are provided in Table 3.11. Remember that the previous value of the horizon length of OGA(b1) in Rb2 in Table 3.8 is $\Delta t^k = 0.144$ s (or $N^k = 25$ instants).

Table 3.11: Cases of study for the horizon length of OGA(b)

Control strategy	Notation	Horizon length	$\bar{\varepsilon}(\mu m)$	Improvement
Classical control	r0	-	14.4	-
OGA(b1)	Hb1	0.024 s (5 instants)	4.7	67.36%r0
	Hb2	0.144 s (25 instants)	4.2	70.83%r0
	Hb3	0.294 s (50 instants)	4.8	66.67%r0
	Hb4	0.594 s (100 instants)	6.2	56.94%r0

Table 3.11 indicates that in general the above four cases of study produce smaller contour error than the reference case r0. The best case is Hb2, which has reduced around 70% the contour error of r0.

Fig. 3.26 shows the effects of the obtained gain profiles on the axis jerk responses and the resulting contour error profiles. In general, from Hb1 to Hb4, when the horizon length is increasing, the gain variations are generally decreasing and therefore the variations of the axis jerk responses are decreasing also.

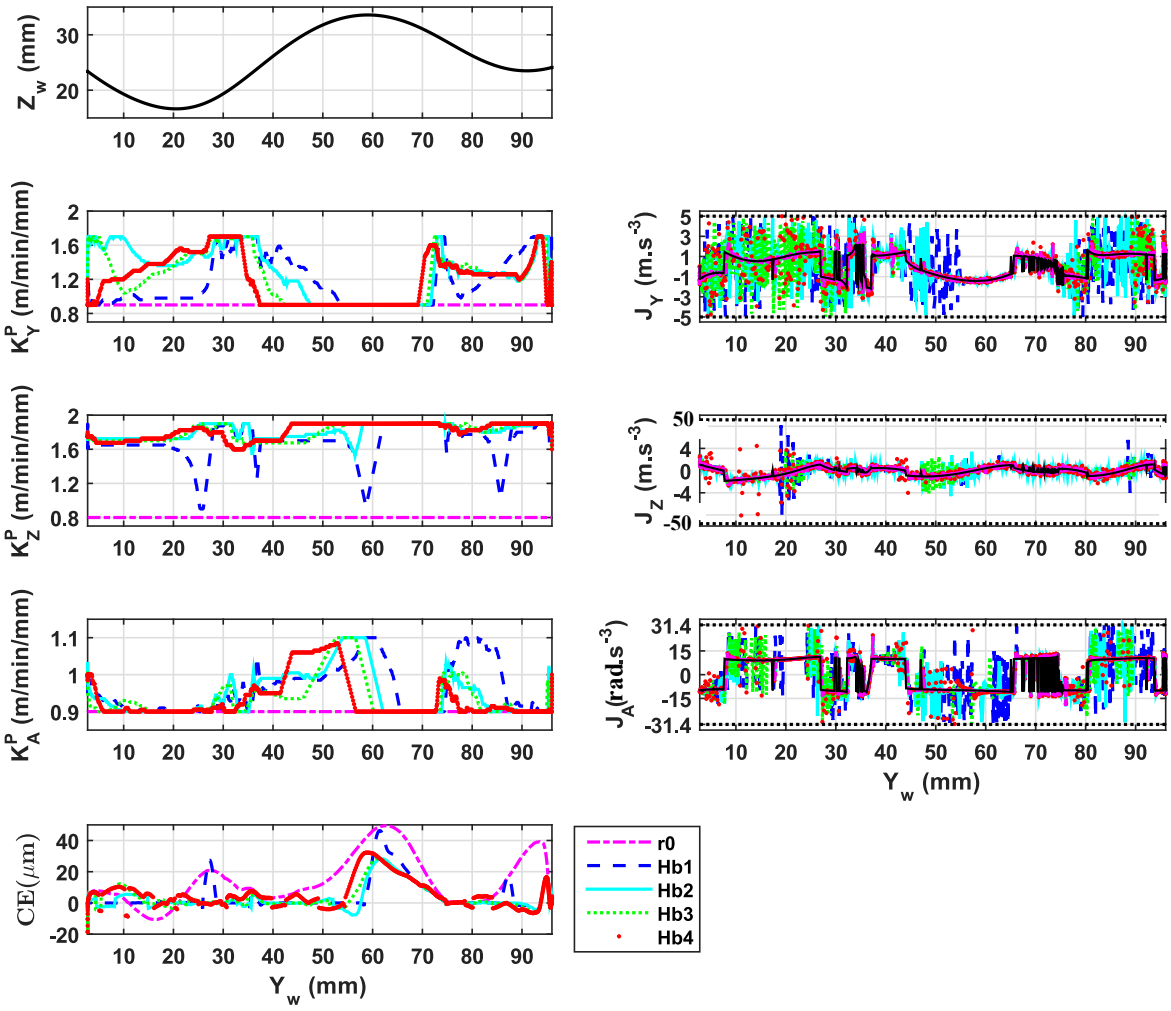

Figure 3.26: Impact of the prediction horizon length on the OGA(b1)

As it can be seen, Hb1 has the largest contour error peaks at the critical positions on the trajectory, being the curvature peaks on the convex and concave portions. Even its large gain variations at these critical positions cannot remove the contour error peaks. This implies that the reactions of the variable gains in Hb1 are too slow to deal with the changes in the curvature at the critical positions, which usually cause large contour errors. This is due to the shortest horizon length of Hb1. Therefore, it can be said that if the horizon length is too short, the OGA(b) will lose the prediction effect at the critical positions on the trajectory.

In Hb4, due to having the longest horizon length, its variable gain profile behaves too early as compared with the other cases, e.g. as seen in the K_Y^P profile at $Y_w = [30 : 50]$ or the K_A^P profile at $Y_w = [50 : 70]$. Besides, it can be noticed that at $Y_w = [50 : 60]$, Hb4 has the largest contour

error in the four cases of study. This can be understood since the detrimental effect of a too long horizon length of OGA(b) discussed in § 2.1, illustrated in Fig. 3.11, that leads to the fact that the obtained variable gain is not optimal for the closest future tool position.

According to the above discussions, the horizon length should be chosen appropriately not only to have the prediction effect, but also to generate the optimal gains. In this way, the horizon length in Hb2 is finally chosen. However, how to choose explicitly the horizon length for OGA(b) is still an open issue. It is proposed that some tests must be performed to choose the best horizon length for OGA(b).

3.8 Impact of the weighting factor

The tuning effects of different values of the weighting factor on the OGA(a) and OGA(b) responses are discussed in this section. The cases of study related to this parameter for OGA(a2) in Ra3 and OGA(b2) in Rb3 from § 3.1 and their quantitative results are given in Table 3.12.

Table 3.12: Cases of study for the weighting factor of OGA(a) and OGA(b)

Control strategy	Notation	λ^F	$\bar{\varepsilon}(\mu m)$	Improvement
Classical control	r0	-	14.4	-
OGA(a2)	Wa1	0	1.4	90.28%r0
	Wa2	0.0001	3	79.17%r0
	Wa3	0.001	5.8	59.72%r0
OGA(b2)	Wb1	0	6.3	56.25%r0
	Wb2	0.0001	4.7	67.36%r0
	Wb3	0.001	4.7	67.36%r0

It can be seen that when the weighting factor λ^F is increasing, the mean contour error of OGA(a2) is increasing; while the mean contour error of OGA(b2) slightly decreases. This can be explained through the fact that OGA with large weighting factor permits only small or maybe too small gain variations. This means that for OGA(a) a small gain amplitude of the pre-defined gain function leads to small gain variations for each horizon, that cannot sufficiently adjust the axis velocity, feedrate, and then cannot much compensate the contour error. In contrast, each optimization cycle of OGA(b) only produces one single optimal gain. Therefore,

small gain variation of each single gain due to large weighting factor does not much affect on its performance. Moreover, similar to the cases of study in § 3.1, for good contouring accuracy, the gain variations of OGA(a) are usually larger than that of OGA(b). Henceforth, the choice of weighting factor depends on the chosen solving technique of OGA, either OGA(a) or OGA(b).

In addition, the effects of the different weighting factors on the obtained gain profiles, the axis jerk responses, and the resulting contour error profiles are illustrated by Fig. 3.27 and Fig. 3.28.

For OGA(a2), when the weighting factor value is zero, as in Wa1, the variable gain profiles undergo larger adjustments, but the axis jerk variations are quite similar to the other cases. Consequently, the contour errors are more reduced.

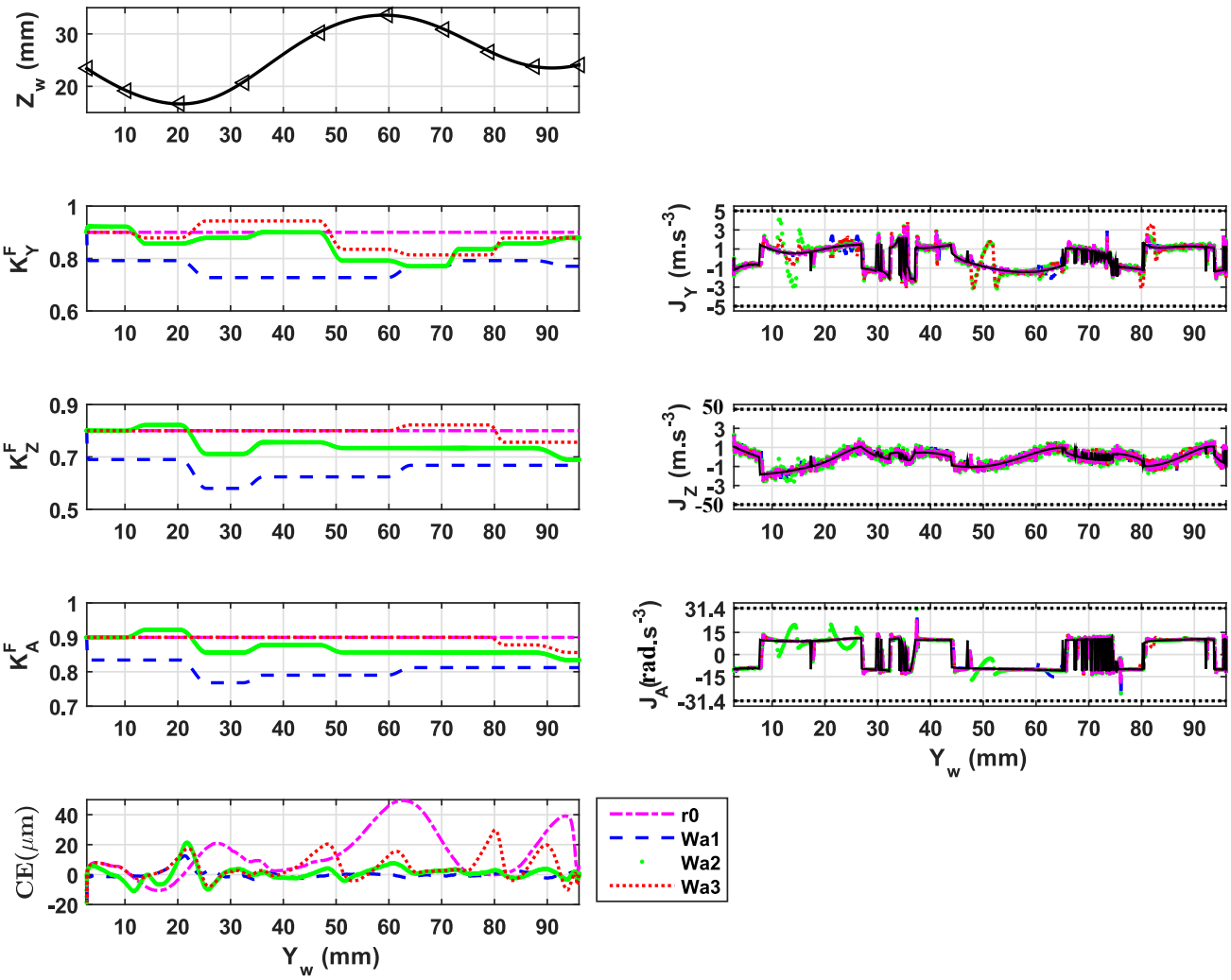

Figure 3.27: Impact of the weighting factor on the OGA(a2)

For OGA(b2), when the weighting factor value is zero, as in Wb1, the variable gain profiles also undergo more adjustments, and the axis jerk responses are more varied than the other cases. Consequently, the optimal gains having large variations may be removed by the OGA constraints, especially the limited jerk constraint, leading to the fact that their contour errors are not much reduced.

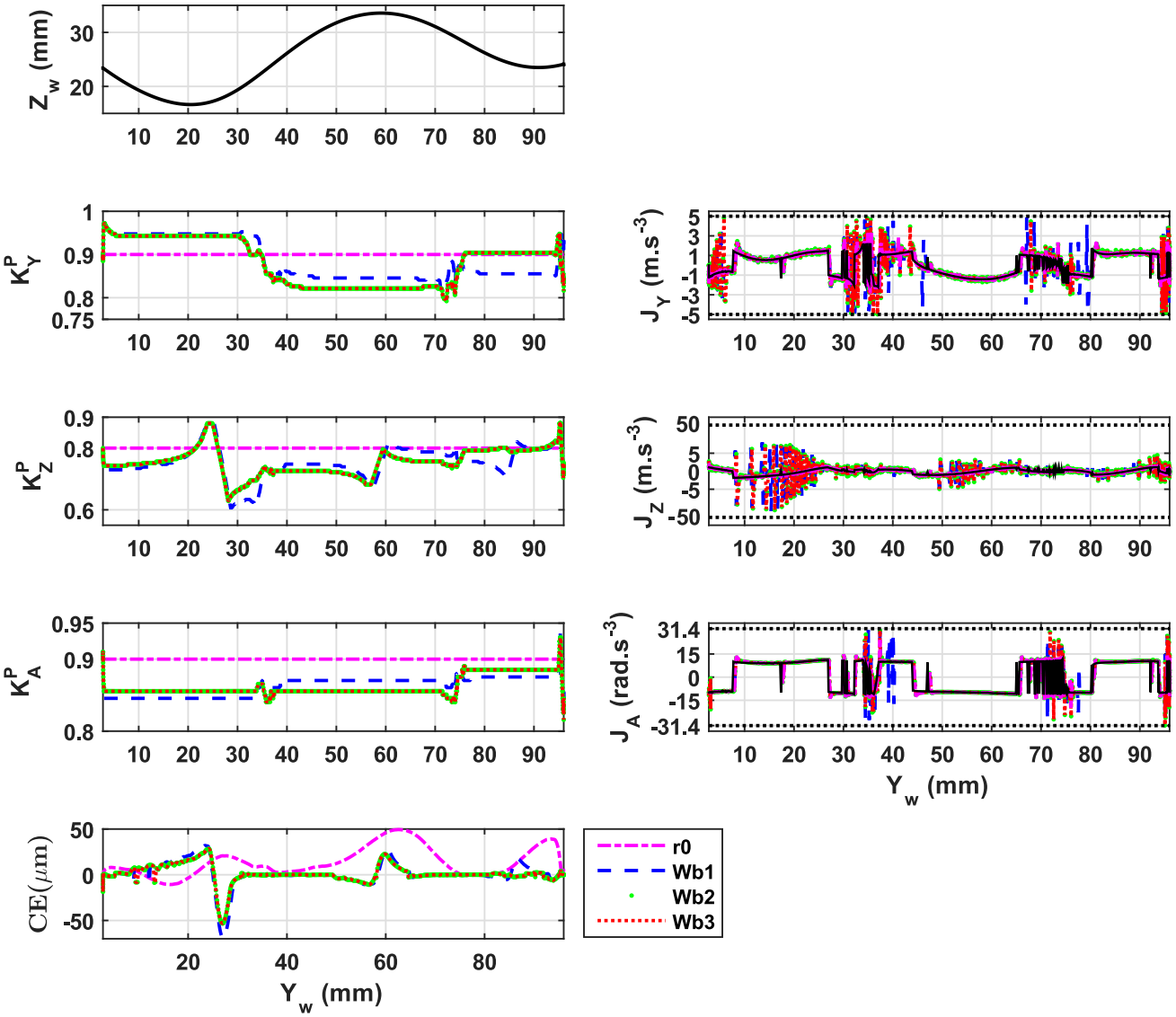

Figure 3.28: Impact of the weighting factor on the OGA(b2)

The above discussions highlight the fact that the weighting factor is a good parameter to regulate the behaviors of not only the variable gain profiles, but also the axis kinematic responses, i.e.

the axis jerk response, and the resulting contour error profile.

4 Conclusion

To conclude, the configuration and the performance of OGA are discussed through the simulation results of the two solving techniques, OGA(a) and OGA(b). Each solving technique has its own advantages and disadvantages or challenges, as follows.

For OGA(a)

The advantages include:

* The adaptive horizon lengths allow OGA(a) to consider the challenges of the different curvature characteristics over the free-form trajectory.
* The adaptive horizon lengths are easily obtained by the Algorithm 1, that can be applied for whatever free-form trajectory.
* The principle of receding horizon without overlap makes the computation time of OGA(a) shorter than OGA(b).
* The behaviors of the variable gain profile and the axis kinematic responses can be managed by the formulation of the gain function, i.e. by choosing an appropriate length of the sinusoidal part.
* The obtained variable gains are shaped into the gain function. They can be computed on-line using the gain function and its optimal parameters. Hence, the optimal variable gains obtained by OGA(a) will need less memory to be stored if OGA(a) is implemented into the real machine.

The challenges include:

* There are many possibilities to define such a gain function of OGA(a). Finding an optimal gain function is an open issue.
* Besides, how to choose the optimal length of the sinusoidal part within the different adaptive horizons is also an open question.

For OGA(b)

The advantages include:

* The prediction effect of OGA(b) is more global than that of OGA(a). It is not necessary to analyze the curvature characteristics of the free-form trajectory before performing OGA(b).
* According to the above results, it can be said that the obtained variable gains in OGA(b) are more flexible than in OGA(a). Moreover, the best cases in OGA(b) produce smaller contour error than in OGA(a).

The challenges include:

* Because the horizon is receded by only one instant, the computation time in OGA(b) is much longer than in OGA(a). It can be argued that if the horizon is receded by more than one instant, then the computation time can be reduced. This is correct. However, initially the OGA(b) is inspired from the idea of MCP, in which one-step receding horizon is chosen. Moreover, if the receding horizon length can be varied, it will be another parameter for tune. Therefore, this will be a part of future works.
* If the variable gains obtained by OGA(b) is implemented into the real machine, it takes more memory space than the gain implementation of OGA(a).
* How to explicitly choose optimal horizon length of OGA(b) is still an open issue.

In addition, the above two solving techniques, OGA(a) and OGA(b), share the same characteristics as follows.

Their common advantages include:

* Both techniques reduce much more contour error than the reference case using the best fixed gain in the classical axis control.
* Considering the optimization over the horizon allows to consider the challenges in the high curvature portions over the free-form trajectory.
* All of the axis kinematic constraints, the stability criterion of the servo drive and the motor current limits are respected.
* They have high possibility to be implemented into the real machine, since the only requirement in the current CNC is the memory to store the optimal gains for the given trajectory and then each gain value is loaded at each sampling time.

- The solving technique is basically based on tuning/searching. The idea is simple, so it can be easily understood and used by new users.

Their common challenges include:

- There are many parameters for choice in the configurations of OGA(a) and OGA(b). How to choose optimal values for some of the user-defined parameters, e.g. the length of the sinusoidal part in OGA(a) or the horizon length in OGA(b), are still open questions.
- The optimizations in the cases of OGA(a3) and OGA(b3) can significantly improve the contouring accuracy, but they have a high computation burden.

According to the above discussions, it can be concluded that if the computation burden can be regardless or overcome within the off-line execution phase, tuning both K_a^P and K_a^F is the best choice for the contouring accuracy. If not only a high contouring accuracy but also a short computation time are needed, then tuning only K_a^F is preferred.

In general, the proposed OGA has significantly improved the contouring accuracy of the given free-form profile, while preserving all of the concerned constraints. It could be a practical solution for the axis control in the future CNC.

Chapter 4

OGA extensions

Contents

The last two chapters have presented, analyzed and proved the core idea of this PhD thesis, that is to keep the classical axis control structure being used in the current CNC and to modify only the control gain inside the position loop by the OGA approach to reduce contour error. In this chapter, the applicable possibilities of OGA are extended.

In the first part, one global control gain is introduced into the classical axis control structure and adjusted by OGA. In the second part, the OGA is also examined in the case using advanced controllers for axis control.

Simulation results are shown and discussed to prove the possibilities of these OGA extensions. Finally, an OGA-GUI interface is created to facilitate the use of OGA.

1 Introduction of the variable global gain

Generally, the OGA is proposed to adjust the control gains used in the position controller (P+FFW) of the classical axis control in the current commercial CNC. This method can be easily implemented by the machine tool manufacturers, because they do not need to change their control structure. The only requirement is to change the way in which the control gain values are stored in CNC, a set of variable gain values instead of a fixed gain value.

This chapter assumes that machine tool manufacturers can develop the axis position control with either classical or advanced controllers. Although the number of control gains or parameters in the position loop can be numerous depending on which kind of controller is used, there is only one reference signal of the velocity loop. Thus, the adjustments of several or all of the control gains result in the modification of the reference signal of the velocity loop, as illustrated in Fig. 4.1.

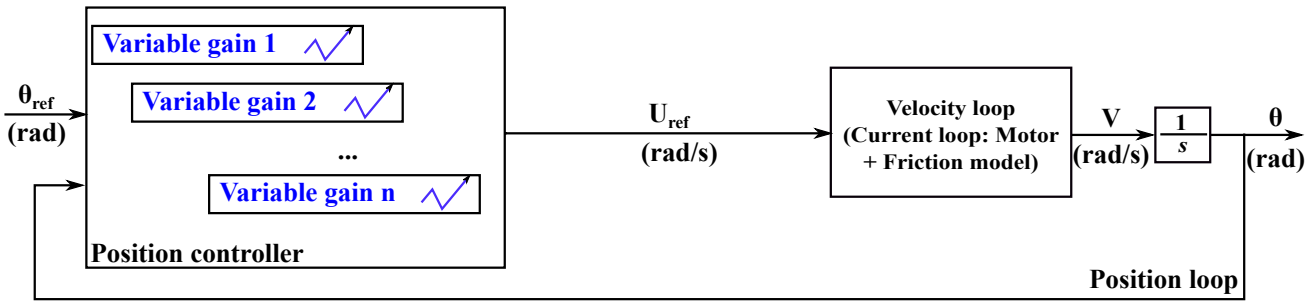

Figure 4.1: Adjustment of the classical control gains in position controller

The more gains adjusted, the more degrees of freedom of the adjustment. However, this leads to a higher computation burden, especially if the optimization is performed for all of the axes. If only one or several control gains are optimized by OGA, it is important to choose the appropriate gains to adjust. The sensitivity and effectiveness of such gains need to be discussed firstly. This is due to the fact that their contribution to the global control action is not the same. For example, as shown in the previous chapter, the velocity reference produced through K^F is much more important than through K^P. Thus, adjusting the K^F gain is more sensitive and more effective than adjusting K^P. However, K^P is essential to maintaining feedback loop stability.

According to the above analysis, to reduce the computation burden and to adequately maintain the efficiency of OGA, a new idea is proposed, in which the control gains inside the position controller, either a classical or advanced one, are kept unchanged, then one variable global gain is introduced just after the position controller, ahead of the velocity loop, as represented in Fig. 4.2. This gain is without unit, as its function is just to modify the amplitude of the reference velocity signal.

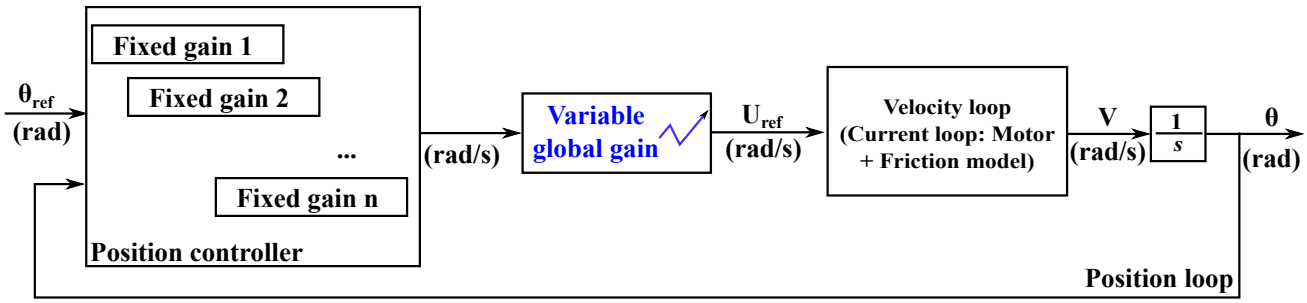

Figure 4.2: Adjustment of the global gain

The OGA is then performed to adjust this new gain. Before analyzing the efficiency of this idea related to the contouring accuracy, it is obvious that the computation time in this case of study is much reduced compared to the global case where all the control gains are optimized, and it is equivalent to the case where only one classical control gain, e.g. either K^P or K^F in the classical axis control, is adjusted.

Another benefit of the global gain adjustment is that it can easily be applied into any type of position controller without regard to the complexity of its control structures. Therefore, CNC technicians or specialists can easily implement this new gain for the application of OGA into their commercial CNC versions.

In the following, such a global gain is namely integrated with the classical axis control structures and with one of the advanced axis controllers.

2 OGA integrated classical axis controller: The global gain case

The above global gain, denoted as K_a^{global}, is firstly added into the classical cascaded control structure through its nonlinear axis model. It is then adjusted by OGA during the off-line execution to produce its optimal variable value to pre-compensate contour error. The axis control structure with the specificity of global gain and OGA is proposed in Fig. 4.3

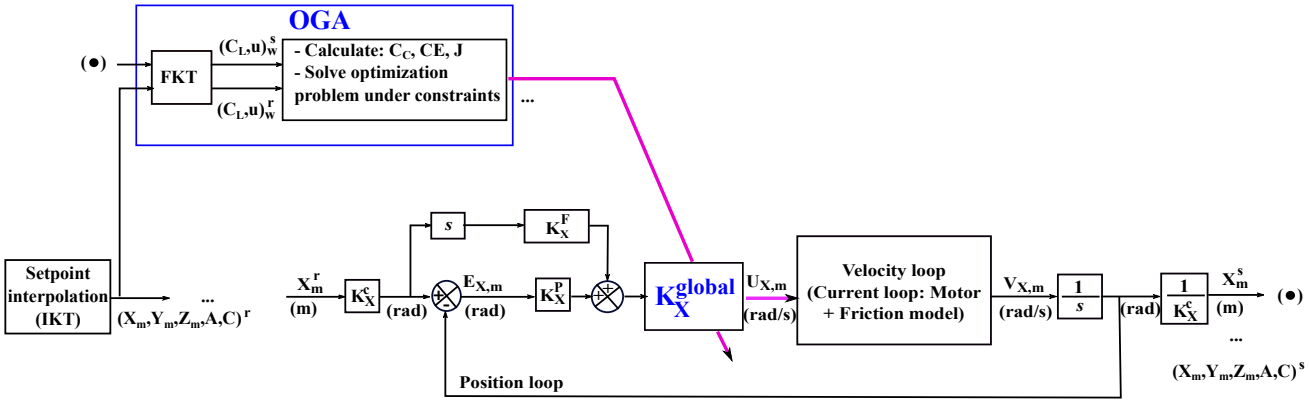

Figure 4.3: Axis control structure with specificities of global gain and OGA

2.1 Admissible range for the global gain

In fact, the principle of OGA formulated in the previous chapter is kept unchanged. The only difference is that the optimization task of OGA is now to optimize the global gain, instead of the P and/or FFW control gains. Therefore, the only requirement is to establish constraints for this new gain.

It can be noticed that the adjustment of K_a^{global} is equivalent to the adjustments of both K_a^P and K_a^F by the same factor. Therefore, the adjusted K_a^{global} values will affect the concerned

constraints proposed in Chapter 2, Section § 3.4, including the stability of the axis drive, the axis kinematic and the motor current limitations. As a result, the bounds of K_a^{global} must be chosen respecting all of the constraints. They are proposed as follows.

Firstly, without the variable global gain in the position loop, K_a^P and K_a^F are chosen as fixed values. In this case of study, it is proposed to choose the two gains as their best fixed values in view of contouring accuracy, K_a^{P*} and K_a^{F*}, given in Table 3.2, and represented by (4.1).

$$\begin{cases} K_a^P = K_a^{P*} \\ K_a^F = K_a^{F*} \end{cases} \tag{4.1}$$

The upper bound of K_a^{global}, $K_a^{global,max}$, is obtained by (4.2).

$$K_a^{global} \leq K_a^{global,max} \quad \text{s.t} \quad \begin{cases} \text{stability criterion proposed in (2.28)} \\ \text{the axis kinematic limitations given in Table 2.2} \\ \text{the motor current limitations in (2.33)} \end{cases} \tag{4.2}$$

In fact, there is a relation between $K_a^{global,max}$ in (4.2), $K_a^{P,max}$ in (2.39) and $K_a^{F,max}$ in (2.42). Assume that $K_a^{global,1}$ is the maximum value of K_a^{global} respecting the stability criterion in (2.28); and $K_a^{global,2}$ is the maximum value of K_a^{global} preserving the axis kinematic limitation in Table 2.2 and the motor current limitations in (2.33). A set of equations in (4.3) must be obtained.

$$\begin{cases} K_a^{global,1} = \dfrac{K_a^{P,max}}{K_a^{P*}} \\ K_a^{global,2} = \dfrac{K_a^{F,max}}{K_a^{F*}} \end{cases} \tag{4.3}$$

Consequently, $K_a^{global,max}$ is in relation with $K_a^{P,max}$ and $K_a^{F,max}$ though (4.4).

$$\begin{cases} K_a^{global,max} & = min\left(K_a^{global,1}, K_a^{global,2}\right) \\ & = min\left(\dfrac{K_a^{P,max}}{K_a^{P*}}, \dfrac{K_a^{F,max}}{K_a^{F*}}\right) \end{cases} \tag{4.4}$$

Then, the lower bound of K_a^{global}, which is obviously greater than zero and less than $K_a^{global,max}$, is chosen by users.

Finally, the admissible range of K_a^{global} is given in (4.5).

$$K_a^{global} \in \left(0 : K_a^{global,max}\right] \tag{4.5}$$

2.2 Simulations and discussions

The applications of the two solving techniques of OGA on K_a^{global} are denoted as OGA(a4) and OGA(b4).

2.2.1 Configurations of OGA

The configurations of OGA(a4) and OGA(b4) are the same as the ones given in Table 3.4, except for the admissible gain range of K_a^{global}. There are many possibilities to choose the $K_a^{global,min}$ (4.5), but only $K_a^{global,min} = 0.5$ is checked considering the reasonable computation time in the simulations. The $K_a^{global,max}$ is obtained according to the discussion in § 2.1. Finally, the admissible range of the global gain is obtained and provided in Table 4.1.

Table 4.1: Admissible range of the global gain

	Y	Z	A
K_a^{global}	[0.5 : 1.2]	[0.5 : 1.5]	[0.5 : 1.68]

The variation factor T_0 in (2.55), which is examined as a tuning parameter, is similar to the cases of study illustrated in Fig. 3.12.

2.2.2 Responses of OGA

The corresponding responses of OGA(a4) and OGA(b4) for each case of study of T_0 are discussed as follows. It should be kept in mind that T_0 in (2.55) is now applied for K_a^{global}, while the two classical control gains, K_a^P and K_a^F, are kept fixed as K_a^{P*} and K_a^{F*}, respectively. The obtained quantitative results are compared with the reference case r0 in Table 4.2.

Table 4.2: Performance of OGA(a4) and OGA(b4)

r0	$\bar{\varepsilon}(\mu m)$			14.4		
		Ra1	Ra2	Ra3	Ra4	Ra5
OGA(a4)	T_0	1	0.2	0.1	0.05	0.01
	$\bar{\varepsilon}(\mu m)$	7.2	3.6	**3.3**	4.2	8.7
	Improvement (%)	50% r0	75% r0	77.08% r0	70.83% r0	39.58% r0
	Computation time (min)			≈ 12.5		
		Rb1	Rb2	Rb3	Rb4	Rb5
OGA(b4)	T_0	0.1	0.05	0.01	0.001	0.0001
	$\bar{\varepsilon}(\mu m)$	25.6	13.1	5.7	**3**	11.7
	Improvement (%)	-77.78% r0	9.03% r0	60.42% r0	79.17% r0	18.75% r0
	Computation time (min)			≈ 400		

It can be seen that the best performances of OGA(a4) and OGA(b4) correspond to the cases of Ra3 and Rb4 where they reduce the contour error of r0 about 77% and 79%, respectively. Although the contouring accuracy of OGA(b4) in Rb4 is slightly improved as compared with OGA(a4) in Ra3, the computation time of the former is 20 times longer than the latter.

To highlight the differences in adjusting the global gain, K^{global}, and other control gains, either K^P or K^F, a comparison between Table 3.8 and Table 4.2 is performed.

It can be seen that the best cases of OGA(a4) in Ra3 is better than the best case of OGA(a1) in Ra1 and worse than the best case of OGA(a2) in Ra3. Similarly, the best cases of OGA(b4) in Rb4 is better than the best case of OGA(b1) in Rb2 and worse than the best case of OGA(b2) in Rb4.

The contouring performances of OGA(a4) and OGA(b4) are respectively worse than OGA(a3) and OGA(b3) in all of the cases of study. This is due to the fact that the former cases have less degrees of freedom in tuning the gains than the latter cases. Because of this, the former cases take much less computation time than the latter cases.

Based on the above comparisons, it can be said that the performances of OGA(a4) and OGA(b4) are more or less the same as OGA(a1) or OGA(a2) and OGA(b1) or OGA(b2), respectively. In addition, although the adjustment of global gain reduces the contour error less than the

adjustment for both classical control gains, the former achieves more reasonable computation time than the latter. Similar to the adjustment of K_a^P and/or K_a^F, the efficiency of the K_a^{global} adjustment much depends on the parameter configurations, such as the gain variation factor T_0.

Fig. 4.4 compares the resulting feedrate, tool tracking error and contour error profiles between OGA(a4) in Ra3 and OGA(b4) in Rb4, in the relation with the trajectory and the obtained variable global gains.

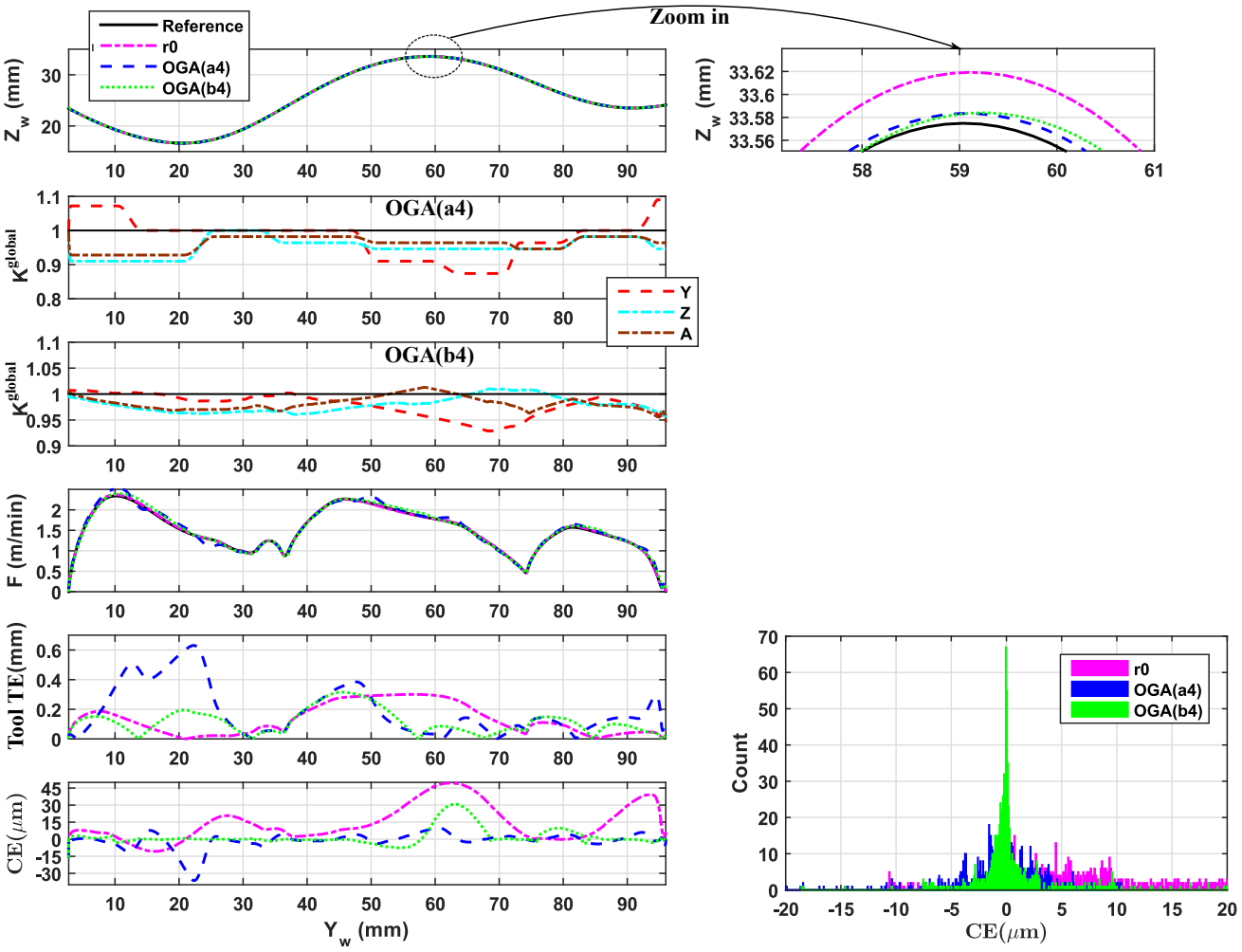

Figure 4.4: Comparisons of feedrate, tool tracking error, contour error between OGA(a4) and OGA(b4)

It can be seen that the obtained variable global gains have slightly modified the response of the feedrate profile, especially on high feedrate zones, leading to the change of the tool tracking error

and the compensation of contour error.

The contour error profiles of OGA(a4) and OGA(b4) have a peak of about 30 μm at the critical regions just over $Y_w = 20\mu m$ and $Y_w = 60\mu m$, respectively. However, in general their contour errors have been much reduced as compared with the one of the reference case.

The contour error histogram shows that the contour errors of OGA(a4) and OGA(b4) have a centered normal distribution, mainly ranging from -10 to 10 μm, while r0 induces the contour errors mainly between 5 to 20 μm.

For OGA(a4), over all of the five cases of study of T_0, it has much improved the contouring accuracy with at least 40 % improvement as compared with the reference case r0. As it can be seen in Table 4.2, it should be good to have the variation factor T_0 ranging from 0.05(Ra4) to 0.2(Ra2) for OGA(a4).

For OGA(b4), from Rb2 to Rb4, the contour error has been reduced as compared with r0, however the mean of contour error in Rb1 is much larger than in r0. This reduced contouring performance of OGA(b4) in Rb1 is due to its largest variation factor, which is explained in detail as follows.

Fig. 4.5 compares the obtained variable global gains and the jerk responses of OGA(b4) in Rb1 ($T_0 = 0.1$) and Rb4 ($T_0 = 0.001$).

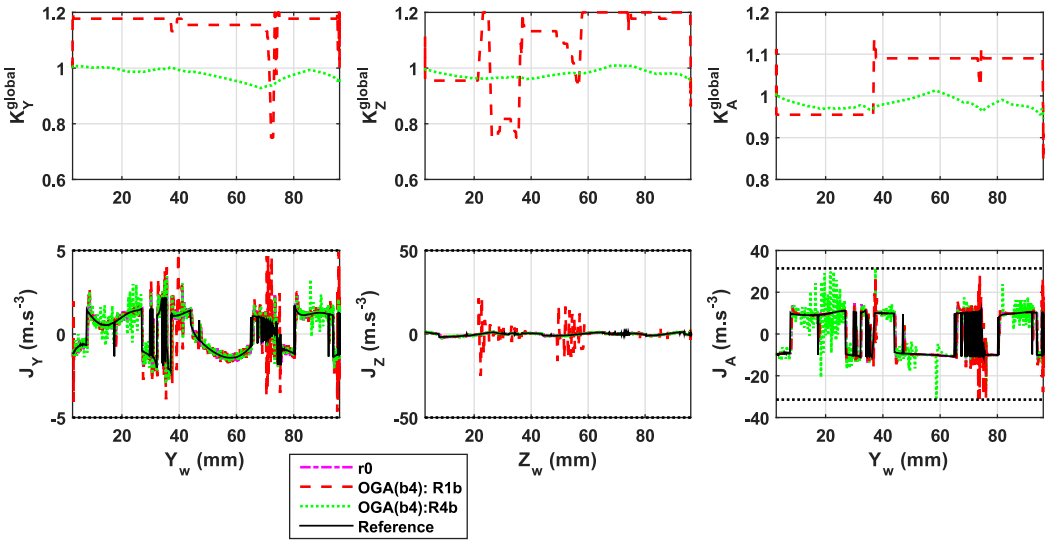

Figure 4.5: Resulting variable global gains and jerk responses of OGA(b4) in the cases of Rb1 and Rb4

It can be seen that the gain profile in Rb4 is much smoother than in Rb1. In fact, the gain profile of Rb1 consists of straight lines and large gain jumps, which are explained by the effect of the kinematic constraints, i.e. the axis jerk limitation, in solving the optimization of OGA. As it can be seen, each gain variation of Rb1 induces a large peak of jerk. Therefore, it can be understandable that if one optimal gain value of Rb1 in term of the smallest contour error corresponds to a large gain variation, that violates the axis jerk limitation, this gain is not chosen by OGA. Consequently, the gain profile of Rb1 remains unchanged in many segments. At the end of the optimization procedure along the trajectory, the obtained variable global gain profile is not an optimal one for the purpose of contour error compensation.

According to the above relation between the gain variation, kinematic constraints and the chosen gain value, the obtained gain profile in Rb1 causes a poor CE profile, as shown in Fig. 4.6.

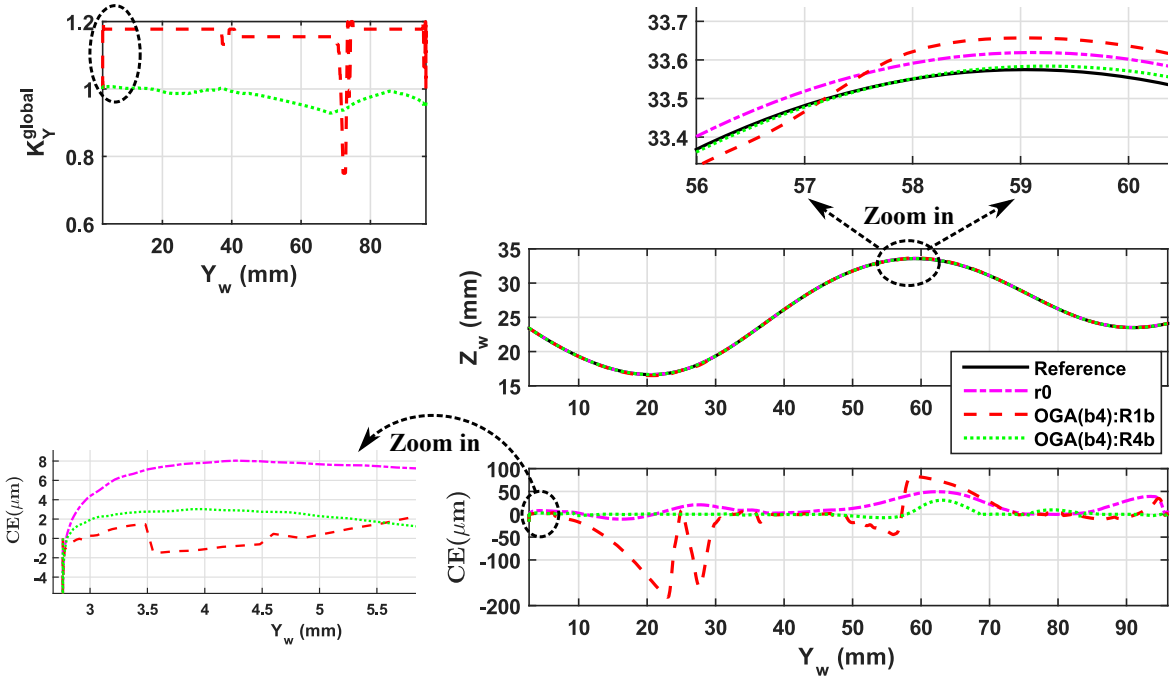

Figure 4.6: Contouring accuracy of OGA(b4) in the cases of Rb1 and Rb4

As it can be seen, at the beginning of the trajectory, OGA(b4) in Rb1 works well when it allows the smallest contour errors within the three cases of study. To obtain this, the gain profile of Rb1 started with large gain variations. For example, the initial variations of K_Y^{global} are as large as 1.2. Then, because the global gain has been increased too much, its next gain variations may violate the jerk limitations. Therefore, after a good start its gain profile kept unchanged until

around $Y_w = 40\ mm$. This explains why the obtained gain profile in Rb1 is not optimal, and consequently the contour error in Rb1 is not reduced, but even increased along the trajectory.

Other analyses of OGA(a4) and OGA(b4) can be performed the same as the ones in previous chapter, such as the impacts of the sinusoidal gain part, of the horizon length or of the weighting factor on the OGA responses(Chapter 3, § 3.6, § 3.7, § 3.8). However, for simplicity, only the axis kinematic responses of both OGA(a4) and OGA(b4) are discussed as follows.

Fig. 4.7 and Fig. 4.8 show the resulting variable global gain profiles and the axis kinematic responses as a function of trajectory in cases of Ra3 and Rb4, which are the two best cases of OGA(a4) and OGA(b4), respectively.

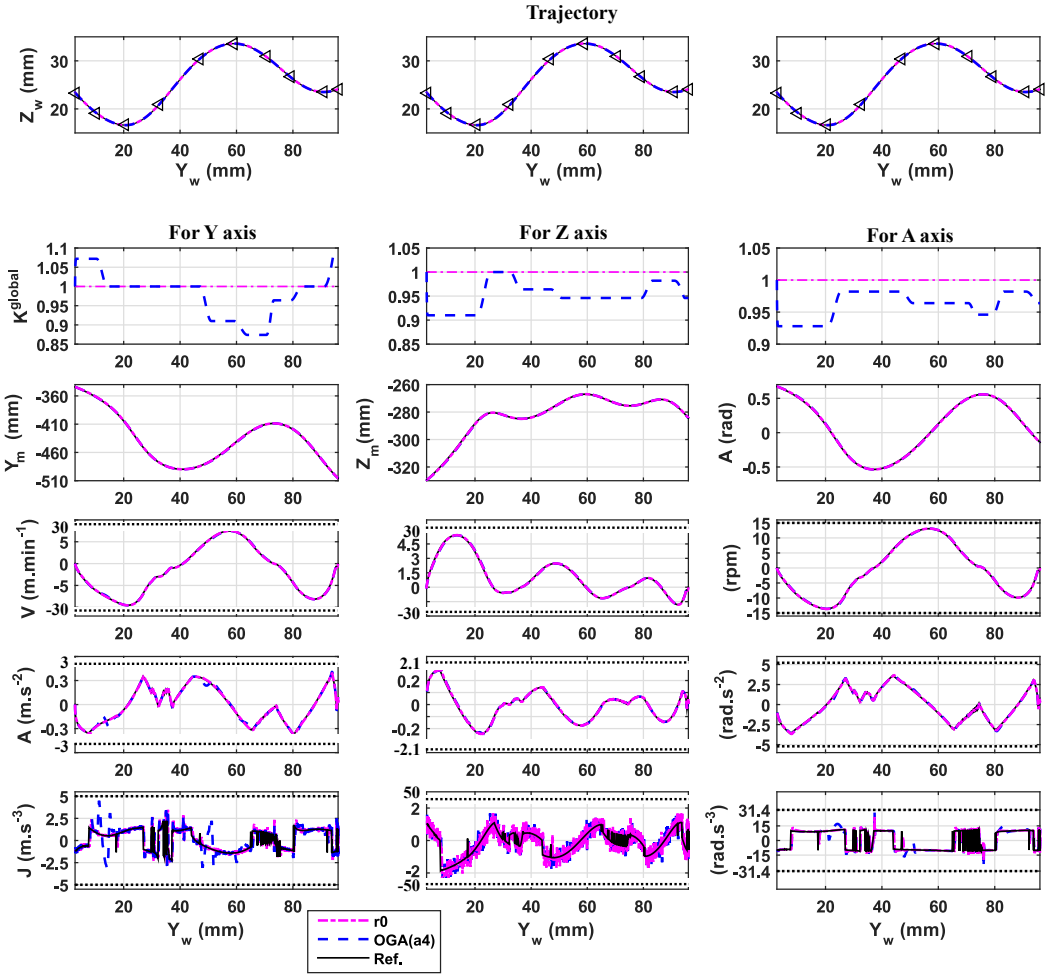

Figure 4.7: Variable gain and kinematic responses of OGA(a4)

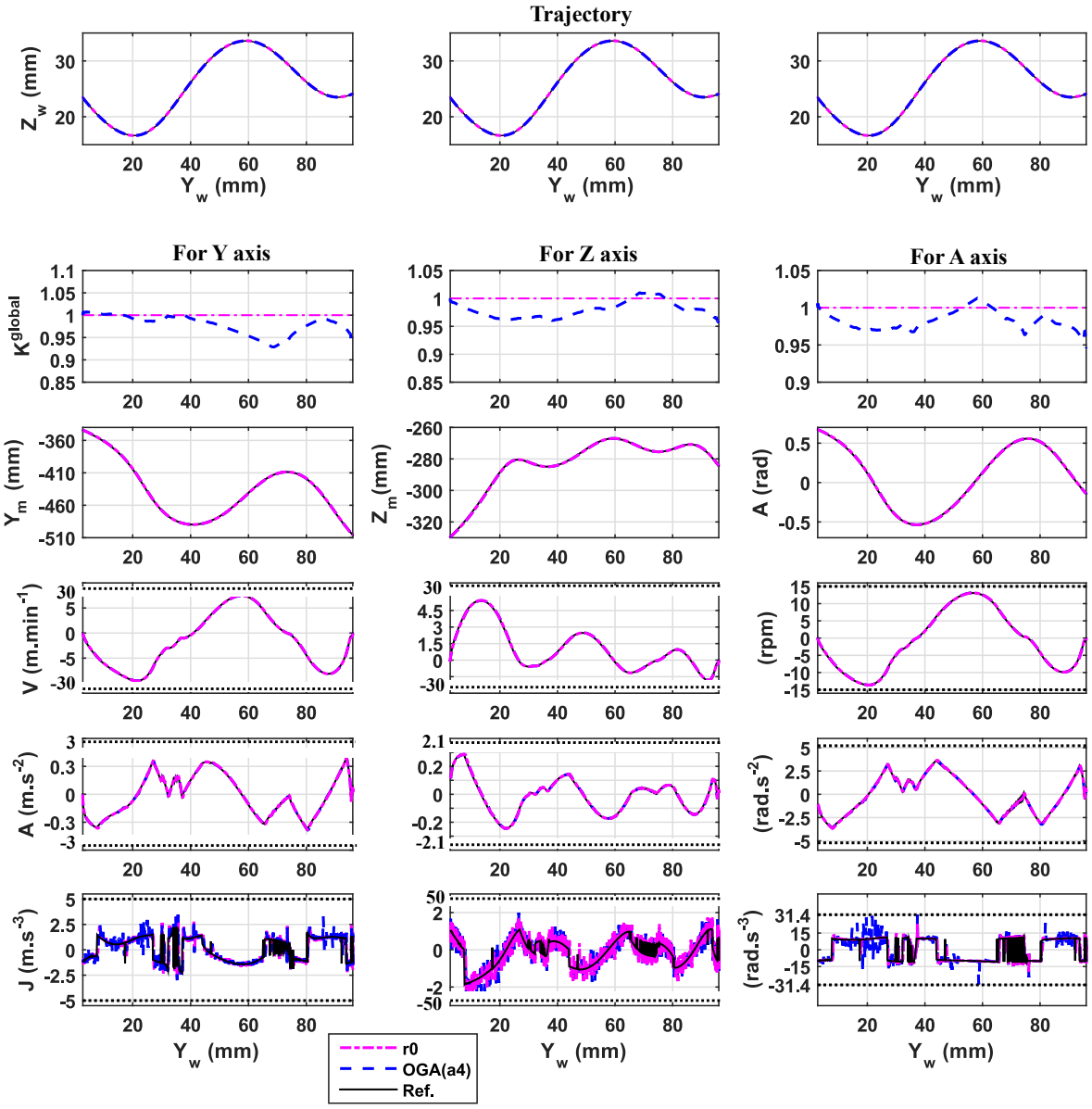

Figure 4.8: Variable gain and kinematic responses of OGA(b4)

As it can be seen, all of the axis kinematic responses respect their limitations in Table 2.2.

It can be seen that in general the amplitude variation of the global gain in OGA(b4) is smaller than in OGA(a4). However, the gain profile of the former varies nearly every instant along the trajectory, while that of the latter has many constant gain segments due to effect of the constant part in its predefined gain function.

2.3 Conclusion

Indeed, it is hard to conclude that the adjustment of the K_a^{global} is more interesting than the previous adjustment of the classical control gains, K_a^P and/or K_a^F. It can only be said that it has the advantages of the one single gain adjustment, especially the reasonable computation time. In addition, the above simulation results prove one important thing that the OGA can be applied correctly as a means of modifying only one single global gain, in the context of the classical axis control structure. Therefore, it is inspired to test this idea with the advanced axis control structure. It makes sense because like in this case the applicable possibilities of OGA can be further extended.

3 OGA integrated advanced axis controller

In this section, the application of OGA for the global gain is examined within the structure of an advanced position controller.

It should be noted that although the development of advanced controller for feed drives is out of the main scope of this PhD thesis, it is also valuable to highlight the generic possibilities of OGA, not only for the classical axis control but also for advanced structures. Moreover, this idea is developed with an assumption that the advanced position controllers can be implemented in the future in a commercial CNC version, which is not obvious.

A case of study for advanced feed drive control is the GPC (Generalized Predictive Control), which has been extensively studied in the literature [Susanu 2006, Dumur 2008, Rodriguez-Ayerbe 2014].

Note that in the previous chapters, the sampling period in the framework of machine tools for the position loop and the setpoint interpolation period were set to 6 ms, that is consistent with the Mikron UCP 710 machining center. Furthermore, the trajectory for the desired part in Fig. 3.1.a is interpolated to machine with two linear axes Y and Z, and one rotary axis A.

Following, the machining of the same part is also simulated. Nevertheless, to evaluate the response of OGA in both different types of position controller and different contexts of multi-axis high speed machining, other configurations for the reference trajectory are proposed, as follows:

- Firstly, the sampling period in the framework of machine tools for the position loop and the setpoint interpolation period were set to 1 ms, instead of 6 ms. The reason for this configuration is that the GPC provides better performance with 1 ms than 6 ms [Rodriguez-Ayerbe 2014]. It can be understood that each advanced position controller has its own specificities, and its own applicable domain. This makes sense, because it allows to see the adaptation of OGA for whatever condition of axis control.
- The above modification related to the sampling period leads to the re-calculation in the feedrate interpolation stage, using VPOp algorithm with a new interpolation period of 1 ms. Unfortunately, this task for the case of 5-axis machining, i.e. for Y, Z, and A axes in the previous case of study, takes too long to finish. However, it can be finished with an reasonable computation time for a 3-axis machining, i.e. for Y and Z axes for this case of study. Due to the time limit in this PhD framework, finally it is proposed that instead of using 5-axis machining center, machining the above desired part with 3-axis machining center is now simulated.

One example for the reference trajectories obtained by the different configurations in CAM and in the feedrate interpolation is illustrated in Fig. 4.9. As it can be seen, the trajectory interpolated with $T_e = 1$ ms has more programmed tool path data than with $T_e = 6$ ms. Furthermore, it is clear that the tool axis orientation of the 3-axis machining is always fixed vertically, while that of 5-axis machining is changed, respecting the configuration of tilt angle $\theta_t = 5$ degrees and yaw angle $\theta_n = 0$ degree, as illustrated in Fig. 1.19.

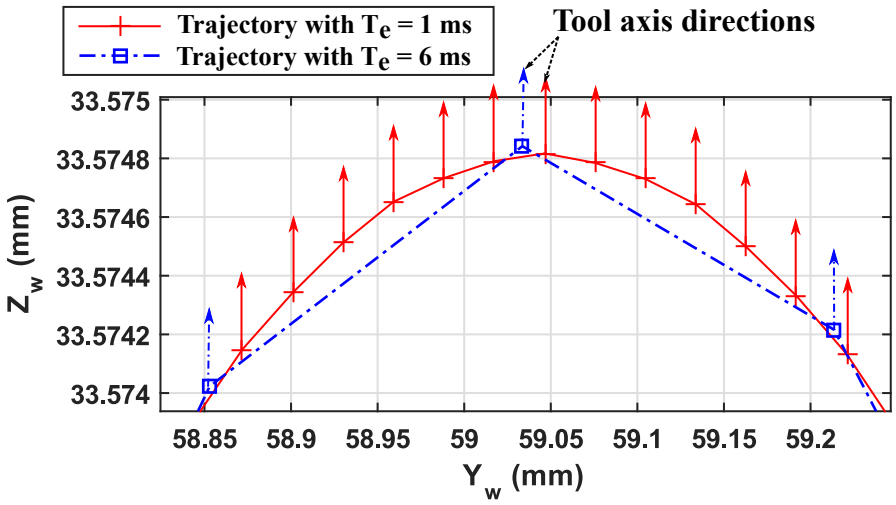

Figure 4.9: 3-axis trajectories with different interpolation periods

In addition, by changing the number of involved axes and the interpolation period in the feedrate interpolation stage, performed by the VPOp algorithm, the kinematic characteristics related to the position setpoints are different. One example is the different reference feedrate cartographies, illustrated in Fig. 4.10.

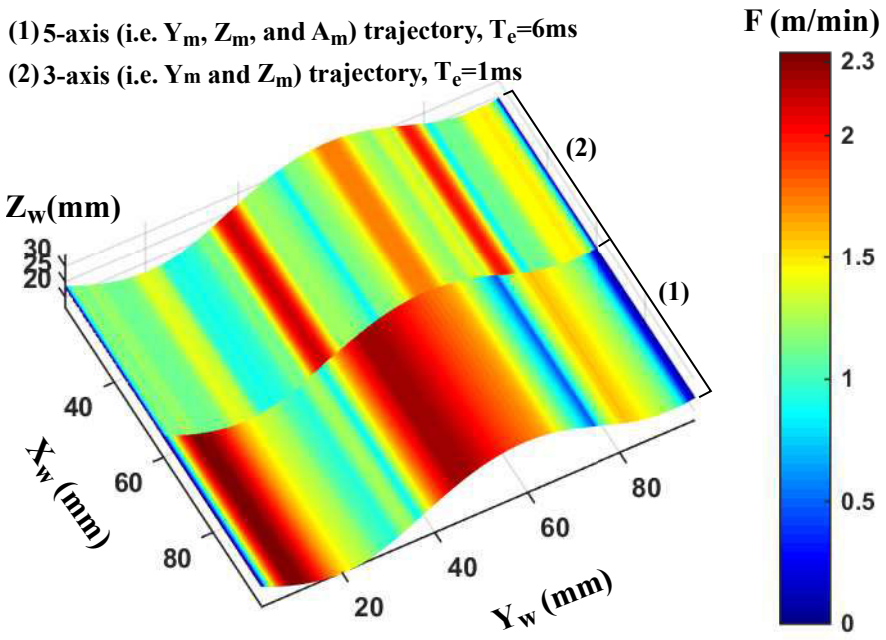

Figure 4.10: Different reference feedrate cartographies for 3-axis and 5-axis trajectories

3.1 Generalized predictive control (GPC)

Following, the synthesis of GPC controller is briefly reminded.

GPC belongs to the class of model-based advanced controllers. Therefore, the design of the GPC controller for the axis control requires the knowledge of a numerical model of the machine axis. In this work, this is obtained through two successive identification procedures as shown in Fig. 4.11.

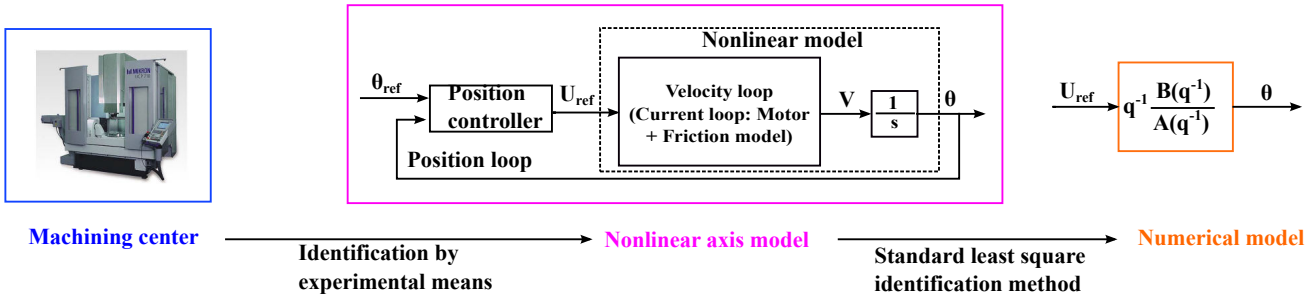

Figure 4.11: Identification procedures of the axis models

The first identification process is to obtain the nonlinear axis model. Based on this nonlinear model, a linear discrete time transfer function of the axis dynamics, in other words the numerical model, is derived through a standard least square identification method [Landau 1990]. Once the appropriate linear axis model is obtained, the design of the GPC can be performed, as stated below.

In the GPC theory [Clarke 1987], the plant is classically modeled by the input/output CARIMA form as in (4.6),

$$A\left(q^{-1}\right) y\left(t\right) = B\left(q^{-1}\right) u\left(t-1\right) + \frac{C\left(q^{-1}\right) \xi\left(t\right)}{\Delta\left(q^{-1}\right)} \tag{4.6}$$

with q^{-1} the backward-shift operator, $u(t)$ the input and $y(t)$ the output of the system. $\xi\left(t\right)$ is a zero mean non-correlated white noise, $C\left(q^{-1}\right)$ models the noise influence [Clarke 1989]. The introduction of the difference operator $\Delta\left(q^{-1}\right) = 1 - q^{-1}$ in the disturbance model helps to find an integral action in the controller and so eliminate the static errors. $A\left(q^{-1}\right)$ and $B\left(q^{-1}\right)$ are the two polynomials obtained by the least square identification:

$$\begin{cases} A\left(q^{-1}\right) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a} \\ B\left(q^{-1}\right) = b_0 + b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b} \end{cases}.$$

The block diagram of the above CARIMA model is represented in Fig. 4.12.

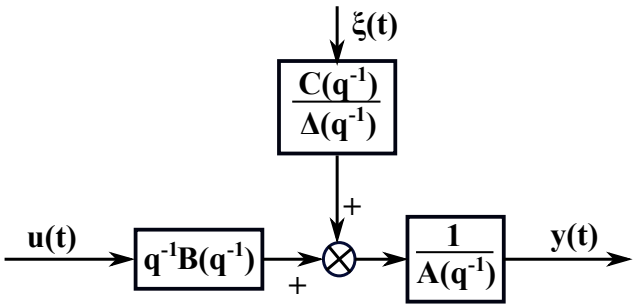

Figure 4.12: Block diagram of CARIMA model

The control signal of GPC is obtained by minimization of a quadratic cost function in (4.7),

$$J = \sum_{j=N_1}^{N_2} \left[y_{ref}\left(t+j\right) - \hat{y}\left(t+j\right) \right]^2 + \lambda \sum_{j=1}^{N_u} \Delta u\left(t+j-1\right)^2 \tag{4.7}$$

in which

- N_1 and N_2 define the lower and upper output prediction horizons, respectively
- N_u is the control horizon,
- λ is the control weighting factor,
- y_{ref} is the setpoint,
- $\hat{y}\left(t+j\right)$ is the optimal j-step ahead predictor, given in (4.8),

$$\hat{y}\left(t+j\right) = \underbrace{F_j\left(q^{-1}\right) y\left(t\right) + H_j\left(q^{-1}\right) \Delta u\left(t-1\right)}_{\text{free response}} + \underbrace{G_j\left(q^{-1}\right) \Delta u\left(t+j-1\right)}_{\text{forced response}} \tag{4.8}$$

  where the F_j, H_j, G_j polynomials are unique solutions of a set of Diophantine equations [Clarke 1987],

- u is the control signal.

The receding horizon principle assumes that only the first value of the optimal control sequence resulting from the minimization of (4.7) is applied to the system, so that at the next sampling period the same procedure is repeated. This control strategy leads to a two-degree of freedom RST controller, implemented through a difference equation (4.9) by means of 3 polynomials R,

S and T, as illustrated in Fig. 4.13. One major characteristic to be noted is that $T(q)$ has a non-causal structure and assures the anticipation effect in the closed loop.

$$\Delta(q^{-1})S\left(q^{-1}\right)u(t) = -R\left(q^{-1}\right)y(t) + T(q)\,y_{ref}(t) \tag{4.9}$$

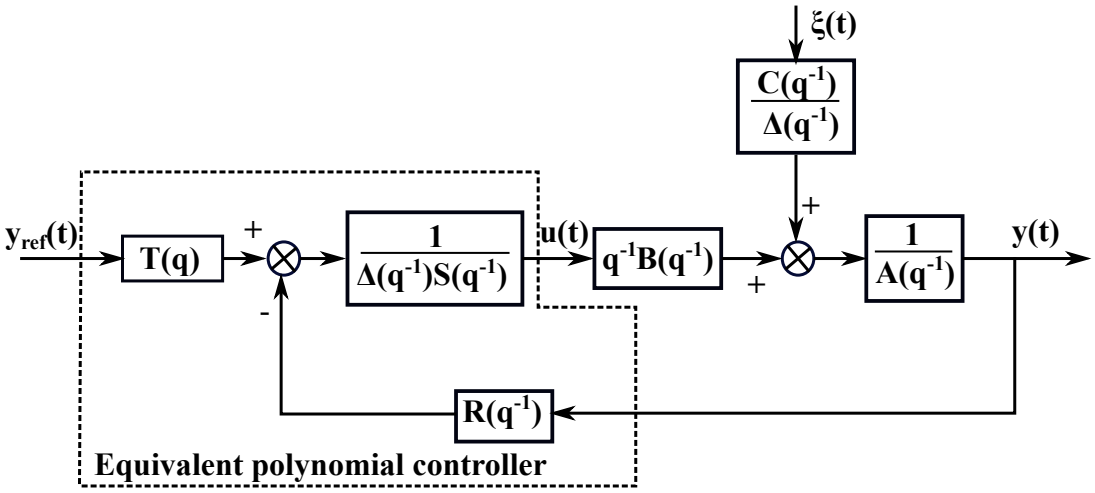

Figure 4.13: Two-degree of freedom RST controller

Once the polynomials $R(q^{-1})$, $S(q^{-1})$ and $T(q)$ of the GPC controller are found, they are used to control the nonlinear axis model, e.g. of Y axis, shown in Fig. 4.14.

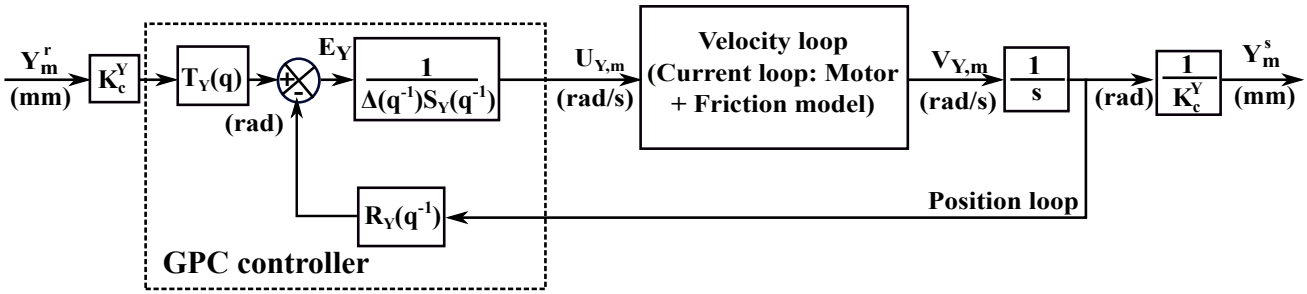

Figure 4.14: GPC controller for the position loop of the nonlinear Y-axis model

3.2 OGA integrated GPC

The global gain OGA strategy is integrated into the above GPC controller through the proposed control structure in Fig. 4.15.

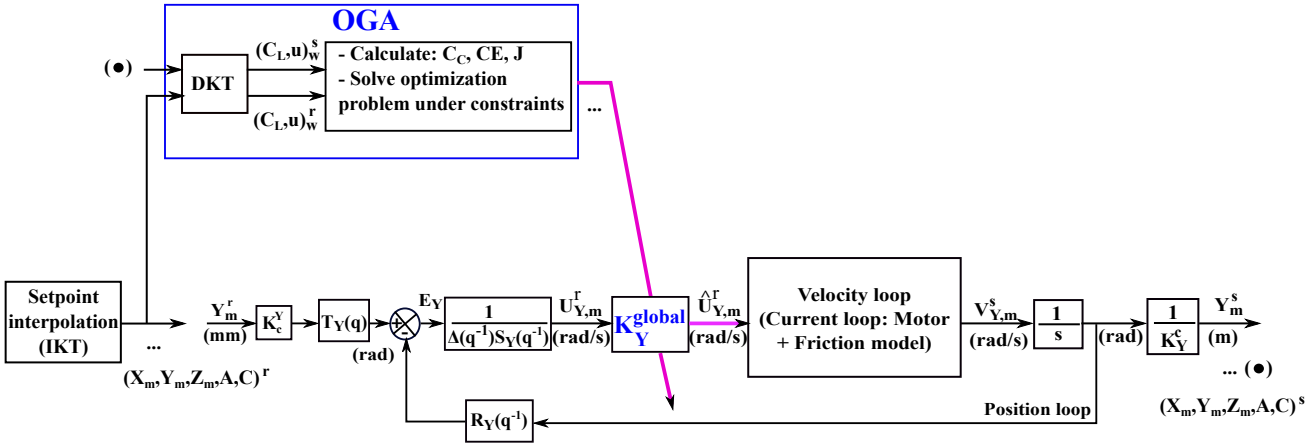

Figure 4.15: OGA integrated RST control structure: the GPC case

The principle of OGA to adjust the K_a^{global} is unchanged, except the change in the upper bound of this control gain.

The upper bound of K^{global} is achieved by respecting two following constraints: (1) tuning it within the position open loop, as illustrated for Y axis in Fig. 4.16, while verifying the stability criterion in (2.28), and (2) tuning it within the position closed loop illustrated in Fig. 4.15, without the OGA, while verifying the axis kinematic limitations in Table 2.2 and the motor current limitations in (2.33).

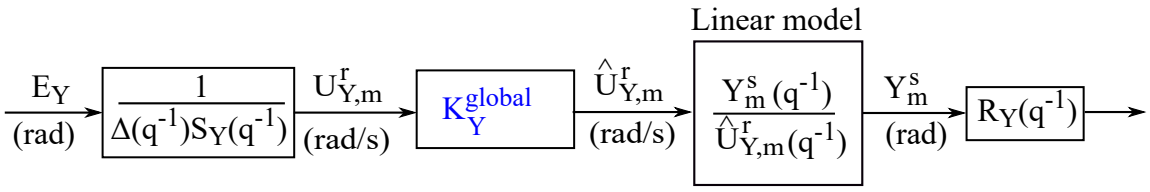

Figure 4.16: Position open loop with GPC controller and linear model

3.3 Simulations and discussions

The main objective of this section is to primarily prove that the OGA can be applied into diversified control structures, either classical or advanced ones and can reduce more or less the contour error as compared with the control structures without OGA.

Due to the above intention, and for the simplicity, only OGA(a4) is performed. Following, four cases of study are considered, as in Table 4.3.

Table 4.3: Cases of study for different axis controllers (T_e = 1 ms)

Control strategy	Notation	Tuning Gain
P+FFW	f1a	None (Fixed gains)
P+FFW+OGA(a4)	f1b	K_a^{global}
GPC	f2a	None (Fixed gains)
GPC+OGA(a4)	f2b	K_a^{global}

3.3.1 Linear model

The linear model for the position loop is needed not only to determine the stability criterion of the axes but also to design the GPC controller. To obtain this, the same identification procedure as Chapter 3, Section § 1.2 is performed, but not represented here for the simplicity. It is noted that because the only difference is the sampling time of 1 ms, instead of 6 ms, a third-order linear velocity model is also obtained. Following, only the obtained mathematical representation of the linear model of the position open loop, illustrated in Fig. 4.16, is given in (4.10).

$$\frac{Y_m^s\left(q^{-1}\right)}{\hat{U}_{Y,m}^r\left(q^{-1}\right)} = \frac{0.000224q^{-1} - 0.0001933q^{-2}}{1 - 2.729q^{-1} + 2.444q^{-2} - 0.6697q^{-3} - 0.04502q^{-4}} \tag{4.10}$$

3.3.2 Reference responses

To perform the cases of study in Table 4.3, the reference cases of P+FFW and GPC controllers must be determined firstly.

Reference response of the classical axis controller (P + FFW)

As discussed in Chapter 3, Section § 1.3, the reference case of the classical axis controller corresponds to the response based on the best fixed gains, K^{P*} and K^{F*}. These gains are obtained by testing many values of K^P and K^F in order to find out the best ones having the smallest mean contour error, respecting all of the constraints.

In this case of study, the above searching approach is employed again to find out K^{P*} and K^{F*}. These best fixed gains and the corresponding mean of contour error are given in Table 4.4

Table 4.4: K^{P*} (m/min/mm), K^{F*} and $\overline{\varepsilon}$ in 3-axis machining with $T_e = 1$ ms

Gain	Y	Z
K^{P*}	1.5	2.3
K^{F*}	0.72	0.57
$\overline{\varepsilon}$ (μm)	0.46	

Comparing Table 3.2 and Table 4.4, it can be noticed that K^{P*} and K^{F*} in 3-axis machining with $T_e = 1$ ms are different from the ones in 5-axis machining with $T_e = 6$ ms. This is due to the fact the position setpoints and its reference kinematic characteristics of these two cases are produced differently by the feedrate interpolation algorithm, as discussed and illustrated in Fig. 4.9 and Fig. 4.10. Therefore, it can be understood that although the trajectory curvature is the same, the control gains need to be consistent with the different kinematic references, i.e. different feedrate profiles, to obtain a high contouring accuracy.

Reference response of the advanced controller (GPC)

The synthesis of GPC polynomials is based on the obtained numerical model of the position loop in (4.10), along with the pre-defined parameters including N_1, N_2, N_u, λ, in (4.7) and the polynomial C in (4.6). These parameters are chosen respecting the stability constraints in (2.28).

In particular, it is proposed that the reference response of GPC corresponds to the best value of λ in (4.7), denoted as λ_a^*, which is achieved by the same manner as the best fixed gains, K^{P*} and K^{F*}, of the P + FFW controller. In this case of study, λ_Y and λ_Z are tuned within a set of values $[0.0001 : 0.0001 : 0.01]$, in order to find out the best values, λ_Y^* and λ_Z^*, corresponding to the smallest mean contour error. Table 4.5 represents the values of all of the predefined parameters in the reference case of the GPC controller, along with its mean of contour error.

Table 4.5: Parameters and $\overline{\varepsilon}$ in the reference response of the GPC controller

Parameter	Value
N_1	1
N_2	12
N_u	1
$C\left(q^{-1}\right)$	$\left(1-q^{-1}\right)\left(1-0.8q^{-1}\right)\left(1-0.9q^{-1}\right)$
λ_Y^*	0.0002
λ_Z^*	0.001
$\overline{\varepsilon}\ (\mu m)$	0.23

The above polynomial C is chosen considering [Rodriguez-Ayerbe 2014], in which the obtained GPC controller for axis control has been experimentally validated. Such a polynomial C has been chosen due to the two following reasons. Firstly, it includes a root at $q = 1$, in order to remove the integral action of the GPC controller. In fact, the static friction in the axis produces oscillations in the output when an integral action is included in the predictive controller [Rodriguez-Ayerbe 2014]. Secondly, the other two roots of the C polynomial permits to obtain good robustness margins [Rodriguez-Ayerbe 2005].

The synthesis of GPC leads to the following R, S and T polynomials, e.g. of Y axis, in (4.11)

$$\begin{cases} R_Y\left(q^{-1}\right) & = 90.8526 - 160.3818q^{-1} + 67.1472q^{-2} + 4.2234q^{-3} \\ \Delta\left(q^{-1}\right) S_Y\left(q^{-1}\right) & = 1.0000 - 1.9246q^{-1} + 1.1778q^{-2} - 0.2262q^{-3} \\ T_Y\left(q\right) & = 0.2534q^{-1} + 0.1279 + 0.0346q^{1} - 0.0475q^{2} - 0.1162q^{3} \\ & -0.1712q^{4} - 0.2124q^{5} - 0.2404q^{6} - 0.2558q^{7} - 0.2596q^{8} \\ & -0.2531q^{9} - 0.2373q^{10} - 14.2170q^{11} + 17.4360q^{12} \end{cases} \tag{4.11}$$

Frequency analysis

Fig. 4.17 shows the Black-Nichols diagrams of the position open loop to verify the stability criterion of the P + FFW and GPC controllers, using the above configurations for the reference cases. For both Y and Z axes, the gain margin and phase margin of the P + FFW controller are larger than the ones of the GPC controller. However, both of them respect the stability constraint in (2.28).

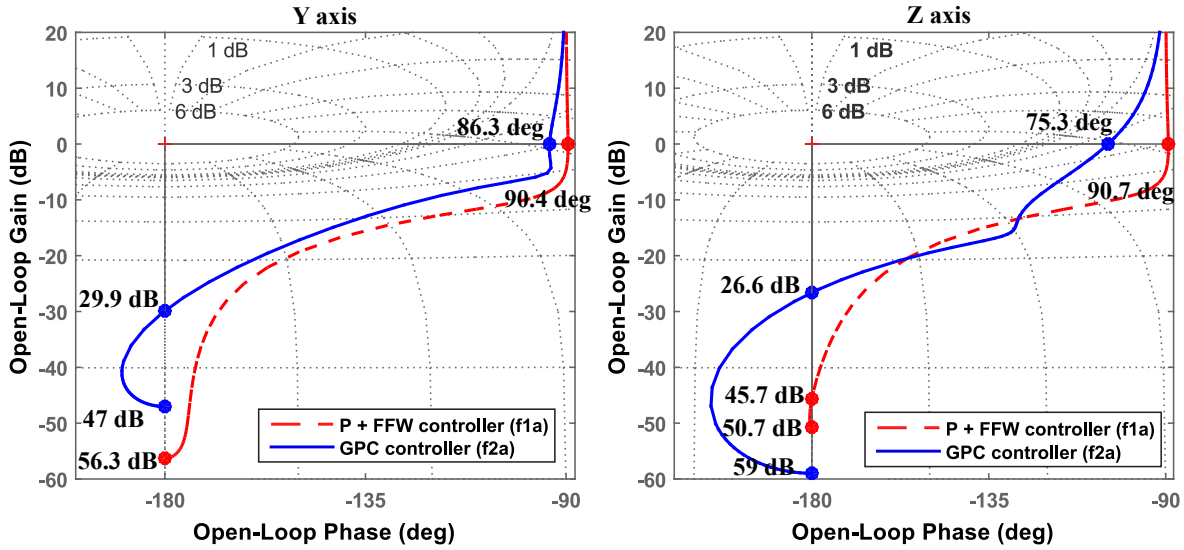

Figure 4.17: Frequency analysis of f1a and f2a: a) Y axis; b) Z axis

3.3.3 Configurations of OGA

For the OGA configurations, the parameters in Table 3.4 are kept unchanged, while the variation factor T_0 is chosen as 1, corresponding to case Ra1 in Fig. 3.12. With the interpolation period of 1 ms, the trajectory of the desired part in Fig. 3.1.a includes 4965 instants. The total length of the tool displacement is $s^{end} = 101.5$ mm. The nine adaptive horizons generated by the Algorithm 1 have the adaptive N^k and $\Delta s^{h,k}$ as follows:

$N^k\,(h1:h9) = [531, 540, 747, 726, 621, 515, 354, 637, 294]$ instant.

$\Delta s^{h,k}\,(h1:h9) = [8.5383, 10.7139, 12.8409, 17.2418, 12.7771, 11.9358, 9.5926, 12.2614, 5.1457]$ mm.

Based on the analysis in § 3.2, the admissible range of the global gain is obtained and given in Table 4.6. Only $K^{global,min} = 0.9$ is checked to have a reasonable computation time, that is about several tenths of minutes.

Table 4.6: Admissible range of the global gain with $T_e = 1$ ms

Gain	Case	Y	Z
K^{global}	r0+OGA(a4)	[0.9 : 1.16]	[0.9 : 1.25]
	GPC+OGA(a4)	[0.9 : 1.62]	[0.9 : 1.24]

3.3.4 Responses of the OGA integrated GPC

The resulting variable global gain profiles and the axis kinematic responses of all of the cases of study in Table 4.3 are firstly shown in Fig. 4.18 and Fig. 4.19.

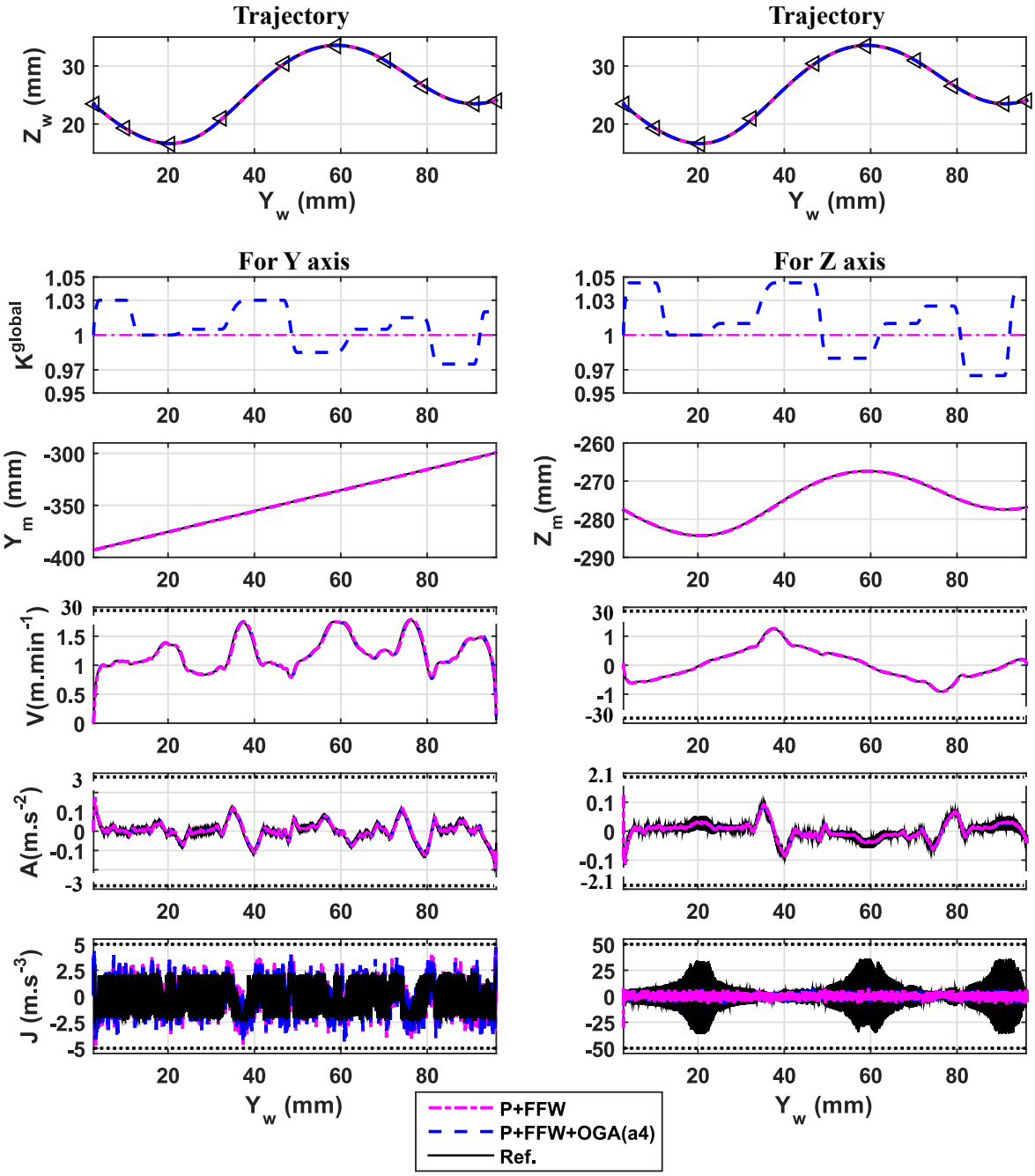

Figure 4.18: Kinematic responses of f1a and f1b

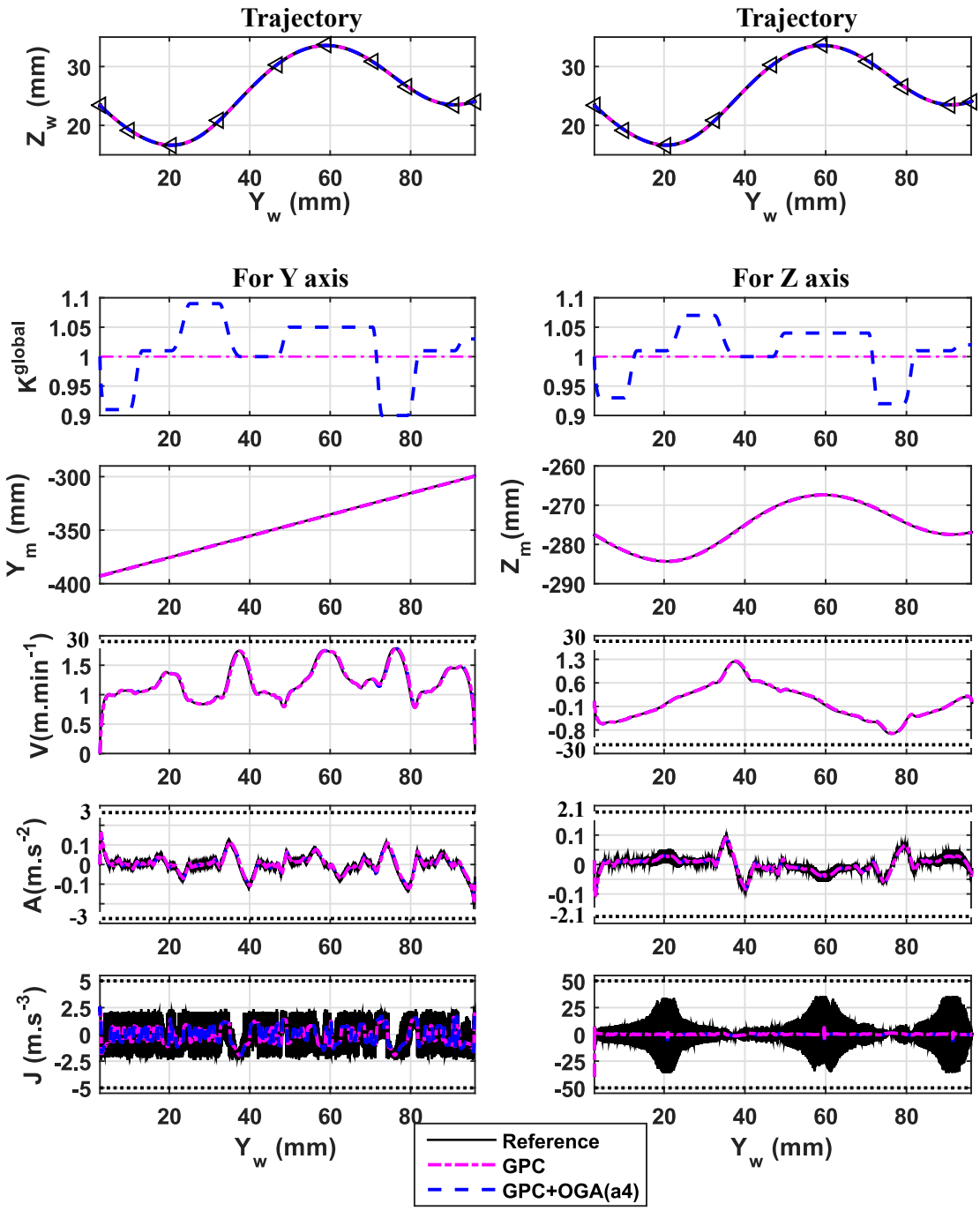

Figure 4.19: Kinematic responses of f2a and f2b

As it can be seen, the global gain is adjusted within its admissible gain range, and all of the axis kinematic responses respect the kinematic limitations in Table 2.2. Moreover, it can be seen that for GPC and GPC +OGA(a4), the variation of jerk response is the least in all of the cases.

The comparison of the mean contour error between the above cases of study is given in Table 4.7. It can be seen that the application of OGA in the two cases f1b and f2b has reduced the contour error in the two reference cases f1a and f2a, with around 17 % and 39%, respectively.

Another remark is that the development of the advanced axis controller, GPC, can reduce 50% the contour error of the classical axis controller, P+FFW. While, if GPC is combined with OGA(a4), then it can even reduce around 70% the contour error of the classical controller.

Table 4.7: Contouring performances comparison

Control strategy	Notation	Mean contour error (μm)	Improvement %	Computation time (min)
P+FFW	f1a	0.46	-	-
P+FFW+OGA(a4)	f1b	0.38	17.39%f1a	≈ 12.5
GPC	f2a	0.23	50%f1a	-
GPC+OGA(a4)	f2b	0.14	69.57%f1a, 39.13%f2a	≈ 12.5

Moreover, Fig. 4.20 illustrates the resulting profiles of the variable global gain, the resulting feedrate, tool tracking error and contour error profiles as a function of trajectory.

It can be seen that due to the optimal adjustment of the global gain value, although the feedrate and the tool tracking error is slightly changed, the contour error has been much compensated.

Furthermore, as it can be seen in the two histograms in Fig. 4.21, the OGA(a4) allows to obtain a centered normal distribution of contour error, ranging from -0.8 to 0.8 μm for P+FFW+OGA(a4) and from -0.2 to 0.2 μm for GPC + OGA(a4). While, without OGA, the norm of contour error ranges from 0.4 to 1.2 μm and from 0.2 to 0.4 μm for P+FFW and GPC, respectively.

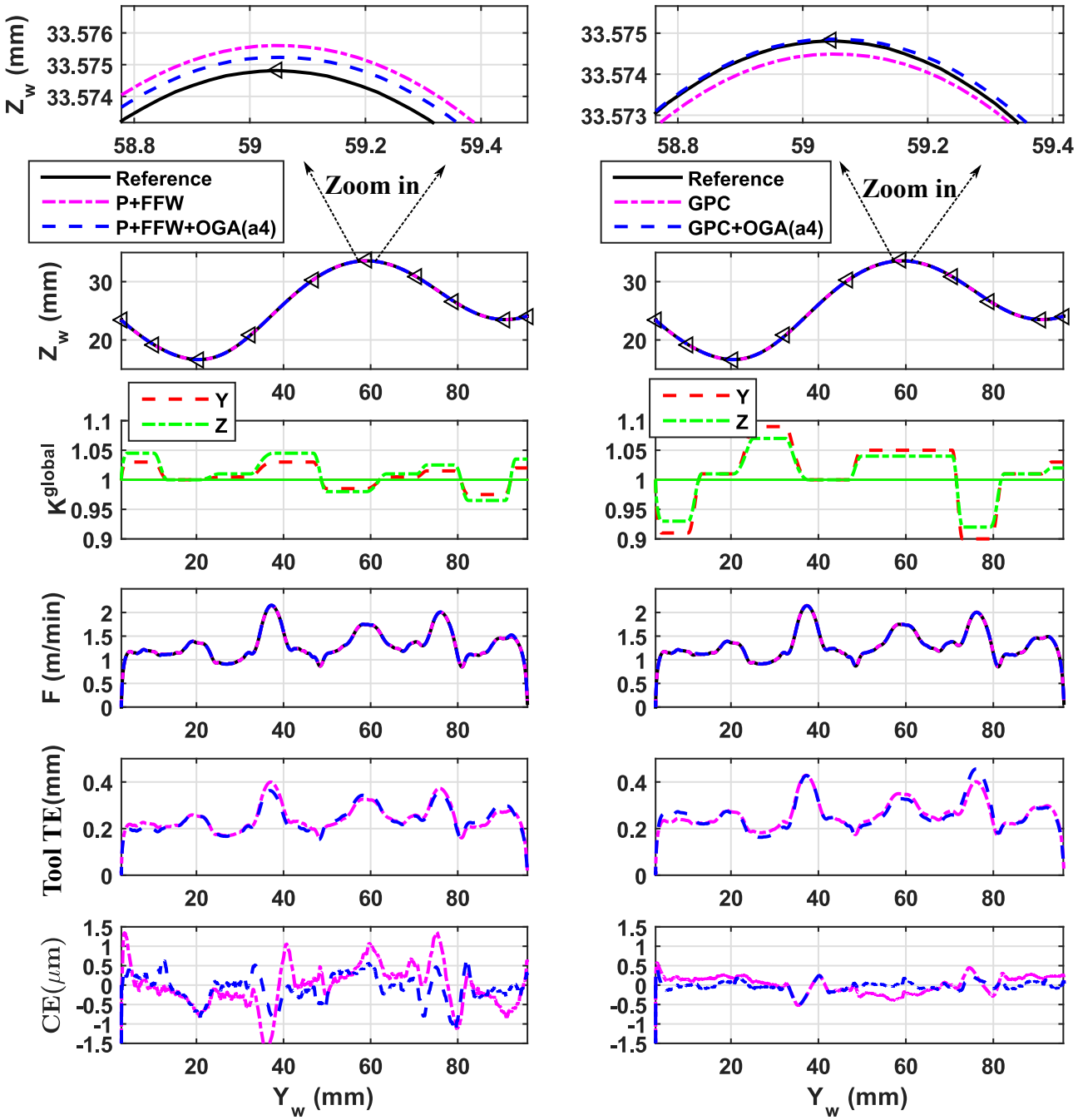

Figure 4.20: Comparisons of feedrate, tool tracking error, contour error between f1a, f1b, f2a and f2b

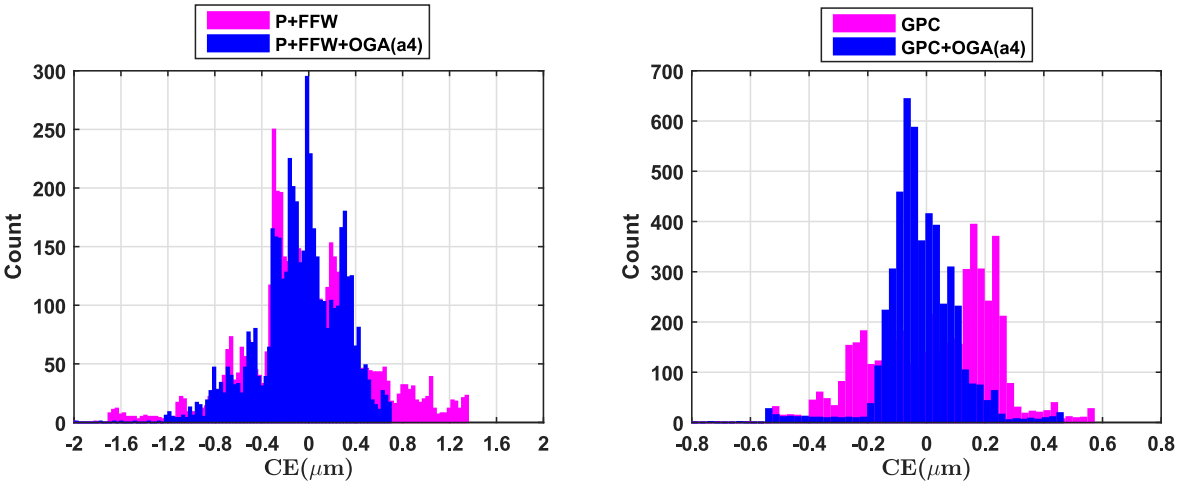

Figure 4.21: Comparison of the histograms of contour error between f1a, f1b, f2a and f2b

3.4 Conclusion

According to the above simulation results and discussions, it can be said that the OGA can be combined with the advanced controller by means of the global gain, in order to much more compensate the contour error.

It is important to remind that the above performances of OGA in these cases of study may not be the best ones, because the parameter tuning is not discussed, it only aims at proving that the OGA can apply and have its effect in different control structures, either classical or advanced one. In this aspect, the above simulation results have satisfied the initial objective.

4 OGA-GUI

To facilitate the end user in configuring and performing OGA, an OGA-GUI interface is developed and illustrated in Fig. 4.22. The working principle of this interface is explained below.

Figure 4.22: OGA-GUI interface

To configure and start OGA, it is necessary to perform 6 following steps:

- Step 1 allows
  - to load the type of machine, e.g. Mikron UCP 710 machine in this work, as illustrated in Fig. 4.23,

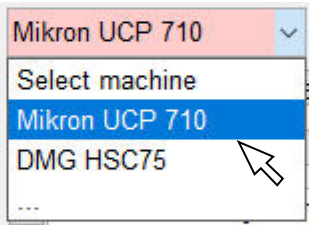

Figure 4.23: OGA-GUI: Select machine structure

– to view the nonlinear axis model (Simulink simulation model), as illustrated in 4.24,

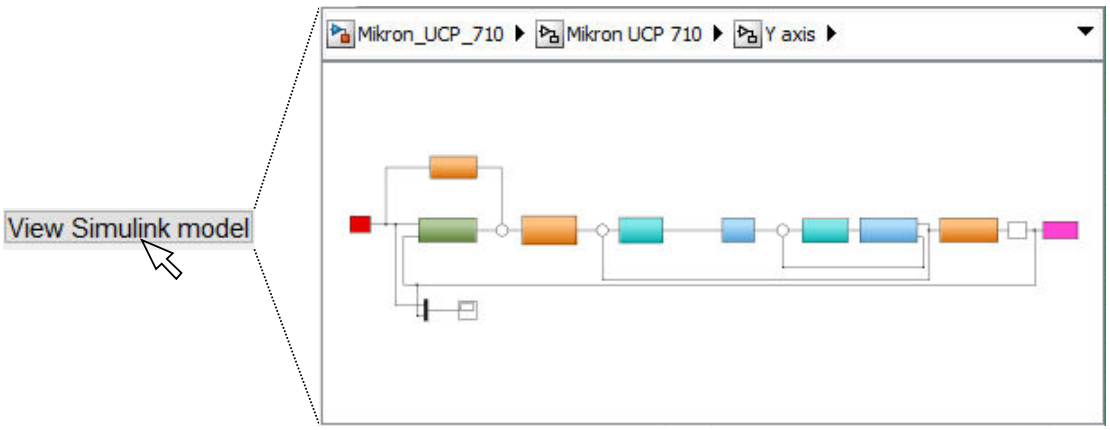

Figure 4.24: OGA-GUI: Simulink simulation model

– to load the parameters corresponding to the machine type and the nonlinear axis model,
– to load the axes setpoints, derived from the feedrate interpolation task. This data is achieved by VPOp algorithm,
– to illustrate the reference surface of the desired part in 3D, and to illustrate the reference axis kinematics, that is the results of the feedrate interpolation stage, as illustrated in Fig. 4.25. The latter allows to verify the reference axis kinematic profile respecting their constraints.

Figure 4.25: OGA-GUI: Reference trajectory and reference axis kinematic

- Step 2 allows
  - to (1) select the type of position controller, either classical controlller (P+FFW) or advanced one (GPC) and (2) set its control gains/parameters, and (3) save these values for the use of OGA, as illustrated in Fig. 4.26,

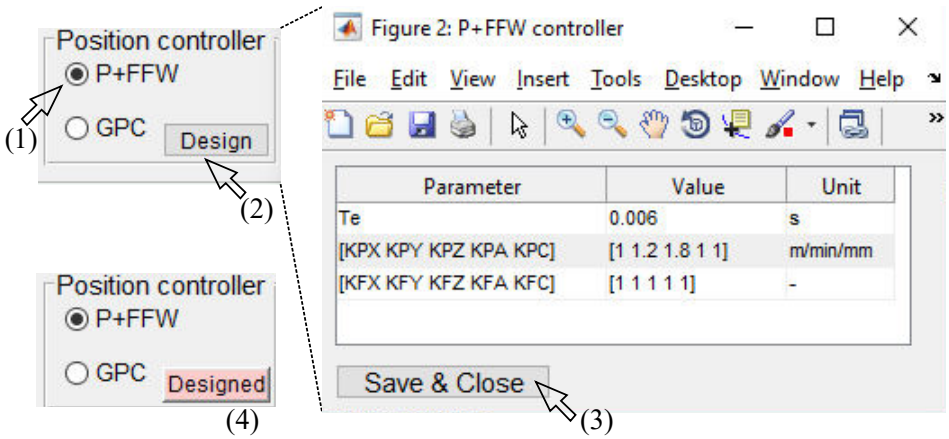

Figure 4.26: OGA-GUI: Select the position controller and design it

  - to select the gain of the position controller to be adjusted by OGA, as illustrated in Fig. 4.27. Note that for the P+ FFW controller, it can be chosen K^P and/or K^F gains; or K^{global}. For GPC, only K^{global} is available to adjust.

Figure 4.27: OGA-GUI: Select adjustable gain(s)

- Step 3 allows to select which axes are involved in the optimization of OGA, either only one axis or several axes or all of the axes. The remaining axes use the fixed gain values for their axis controllers, as illustrated in Fig. 4.28.

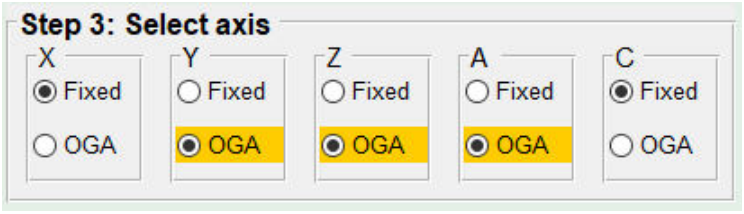

Figure 4.28: OGA-GUI: Select the axes for optimization by OGA

- Step 4 allows

  – to (1) select the solving technique of OGA, either OGA(a) or OGA(b), and (2) configure the specificities of OGA(a) or OGA(b), and (3) save these configurations, as illustrated in Fig. 4.29,

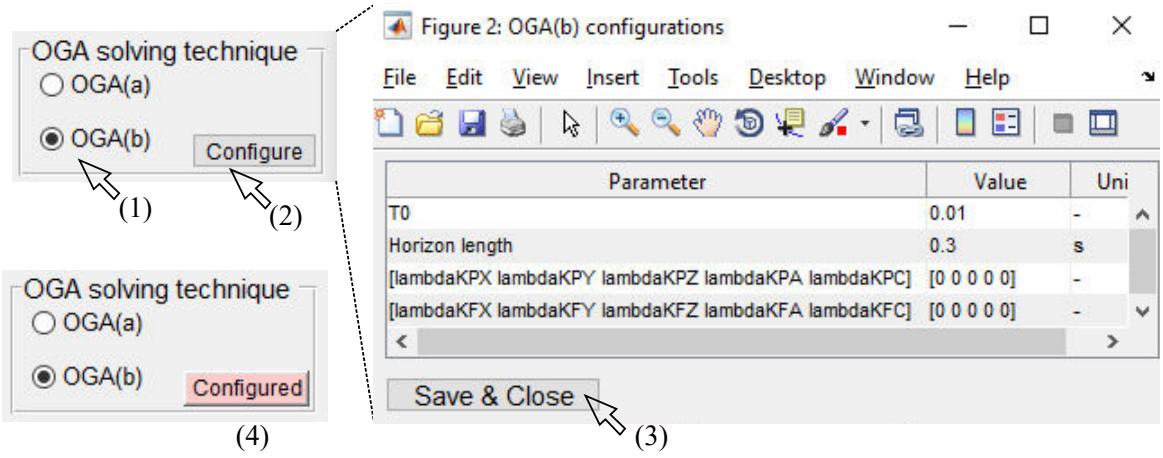

Figure 4.29: OGA-GUI: Select OGA(a) or OGA(b) and configure it

  – to add the values of the constraints, including the axis kinematic limitations, the admissible gain range and the motor current limits, as illustrated in Fig. 4.30.

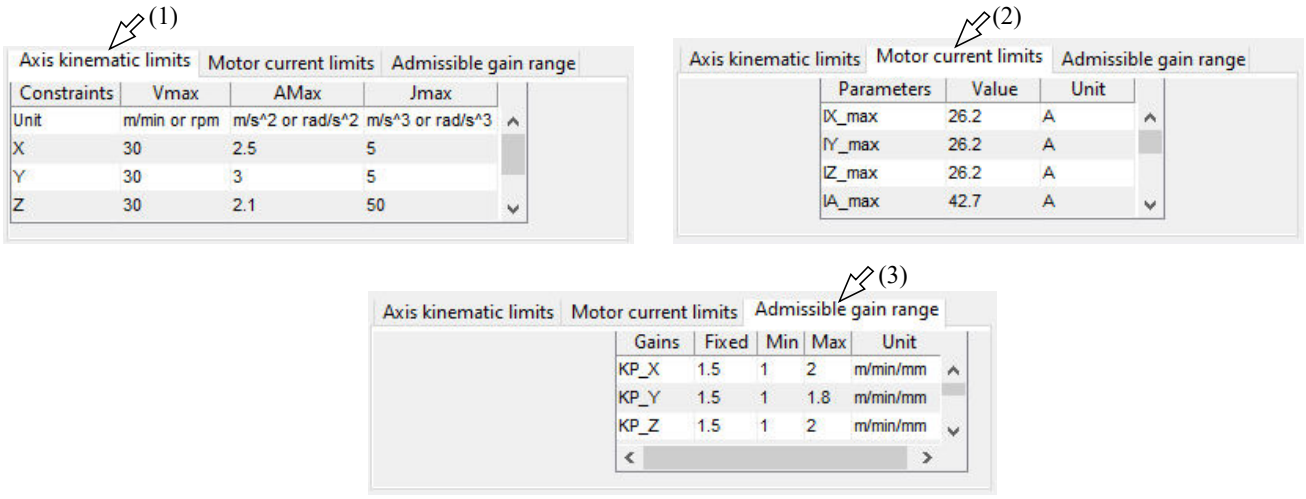

Figure 4.30: OGA-GUI: Add the constraints

- Step 5: After the configuration is done, the user clicks the button: Run OGA, to start the OGA algorithm, as illustrated in Fig. 4.31.

Figure 4.31: OGA-GUI: Run OGA

- Step 6: When the OGA is finished, this step allows to view the resulting variable gain profile, the resulting feedrate and the obtained contour error profile, and to load and save the obtained results, as illustrated in Fig. 4.32.

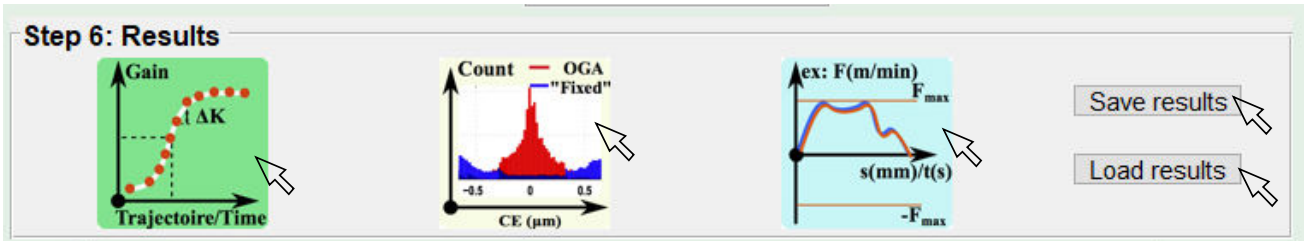

Figure 4.32: OGA-GUI: OGA results

5 Conclusions

This chapter has introduced a new idea related to the global gain and the application of OGA into this new control gain. The advantages and disadvantages of this proposed idea are given as follows.

- Advantages

  The application of OGA for the global gain can lead to a high contouring accuracy, and the same computation time as when the OGA is applied into one single control gain K^P or K^F.

  The general principle of OGA is not changed in this case of study. Only the admissible range for this new control gain is needed to define, but the idea and the constraints for such an admissible range is not changed.

  OGA can be applied not only with a classical controller but also with an advanced one, e.g. the GPC controller in this case of study.

- Disadvantages

  A drawback of this method is that it is needed to slightly modify the software part in CNC, in order to implement this new control gain.

Although the synthesis of the advanced controller is out of the scope of this PhD thesis, it is also interesting to see that the GPC, in the framework of the advanced model-based controller, can give a better contouring accuracy than the classical axis controller for the position loop in CNC.

The OGA-GUI interface is also built to facilitate the end users to tune the parameters, to switch between the techniques and the control gains, and to start the OGA algorithm. The visual comparisons and analysis are also integrated in this interface. Finally, the user can easily achieve the resulting variable gain profiles for use in the future CNC, to pre-compensate the contour error in the machining process.

Conclusion and perspectives

Conclusions

This PhD thesis focuses on improving the contouring accuracy in machining free-form part in the context of multi-axis high speed machining, for either 3-axis or 5-axis high speed machining center, based on the means of an off-line process. Its main strategy is to exploit all possibilities of the classical control structure of axis drive, proposing the elementary base of a method that could be considered for an implementation in a commercial CNC.

Inspiring from the above motivations, the main contributions of this thesis are summarized as follows.

The first contribution is related to the contour error formulation in 5-axis point milling, based on ε_C, using the proposed estimation approach for the tool contact points $\mathbf{C_C}$. In the literature, almost all of works proposed the contour error estimation approach based on ε_L and/or $(\varepsilon_L, \varepsilon_\mathrm{u})$, using the tool path information $(\mathbf{C_L}, \mathbf{u})$. This is because this information is available and can be accessed during machining process, while the determination of the tool contact points $\mathbf{C_C}$ during machining stage is still an open issue. In fact, the use of ε_L and/or $(\varepsilon_L, \varepsilon_\mathrm{u})$ is to estimate the effective cutting error (under cut or over cut), expressed by ε_C. Therefore, the formulation of contour error by ε_C, using directly the estimated tool contact points $\mathbf{C_C}$, is proposed to predict the effective cutting error more accurate than the one based on the tool path information, ε_L and/or $(\varepsilon_L, \varepsilon_\mathrm{u})$.

In addition to this, the kinematics taken into account during in the feedrate interpolation stage have also been analyzed. They are the essential elements supporting generally the axis control and particularly fulfilled the development of the proposed adaptive control law. Especially, the axis kinematic constraints, including the axis velocity, acceleration, and jerk limitations, in the feedrate interpolation are used as one of the constraints in solving the optimization problem of the proposed axis control approach.

The thesis has scoped with the challenges in machining with high speed machining center, being the high feedrate and the axis kinematic constraints of the machine. Besides, the nonlinear characteristic derived from the static friction is also considered through the friction model. The motor current limits have also been taken into account to assure a good behavior of the machine drives. Moreover, the closed structure of commercial CNC and axis drives have been also discussed and taken into account, in order to proposed an easily implementable solution in

actual CNC systems.

An adaptive axis control law has been proposed for the commercial CNC based on an Off-line Gain Adjustment (OGA) approach, to compensate the contour error during machining. The main idea is to determine the optimal variable gain values used in the position loop of the classical axis control to reduce the contour error during the machining phase. This optimal gain calculation is done off-line by solving an optimization problem using a machine virtual environment. The optimization is formulated subject to the axis kinematic constraints, the stability criterion of servo drive, and the limits of motor current. The optimization result is the gain values of the position loop and feed forward for the considered machining profile. In this way the gains of the position controller and the feed forward are changed on-line in real time to reduce the contour error in the machining.

One of the important contributions of the current research is the development of two solving techniques for the optimization of OGA. They are based on the receding horizon principle. The gain tuning is performed within a trajectory horizon, that is receded until the end of the trajectory.

In the first solving technique, OGA(a), the gain is obtained based on the pre-defined geometry horizon, considering the curvature characteristic of the free-form profile and based on the pre-defined gain function. When the horizon is receded without overlap, the optimization is repeated for the next horizon.

In contrast, the second solving technique, OGA(b), predicts the optimal constant gains over one horizon length but only the first optimal gain is kept and the horizon is receded one step ahead. This solving technique has a more important computation burden than the first one but allows the control gain to vary more flexibly. That leads to fine-tune the resulting feedrate and as a consequence, this second solving technique allows to reduce the contour error much more than the first one. In the concerned cases of study, the best case of OGA(b) has improved 67.14% the contouring accuracy, compared with the best one of OGA(a). However, the former takes more than 9 hours to finish the OGA optimization than the later.

The two solving techniques are different in their own principles therefore their configurations are also set differently. The analysis of the parameter configuration on the performance of OGA have been discussed. The choices of the maximum number of potential gains, the gain variation factor, the horizon lengths and the weighting factor are important for the high performance of

OGA.

Another contribution of this work is the proposition of a global variable gain in the axis controller. The influence of the position control gain variation and the feed forward gain variation in the contour error reduction has induced this new idea related to a global control gain. This gain can be integrated into whatever kinds of controllers, either classical or advanced ones. Afterwards, this new gain is used in the application of the proposed OGA approach. This method guarantees not only the improvement of the contouring accuracy but also a reasonable computation time. That extents the tractable possibilities of the proposed OGA approach in an industrial environment.

Simulation results based on a validated machining simulator showed that the precision level of the contour following and the computation time are different depending on the different solving techniques and the number of axes and control gains involved in the optimization of OGA. The results also verified that the responses of OGA have respected all of the concerned constraints. The relations between the obtained variable gains, axis jerk response, resulting feedrate and contour error profiles have been highlighted. Due to the high confidence of the machine simulator, it can be said that the contour error has been reduced significantly thanks to the proposed OGA approach.

In addition, a machine interface has been built (OGA-GUI) to make the use of OGA easier to users and increase the practical applications of the OGA in the machining community.

It can be concluded that the proposed OGA approach satisfies the initial objective of the PhD thesis, that is to reduce the contour error while keeping unchanged the classical axis control structure inside the current commercial CNC of machine tools. The only proposed modification on the future CNC is the extended functions, in which the gains are updated at each sampling time and the required memory to store the gain values for a given trajectory.

Perspectives

In perspective, the OGA will consider more nonlinear characteristics and disturbances effects, e.g. machine flexibilities, into the optimization problem under constraints of OGA. That can extend the effectiveness of the OGA.

The application of OGA for the multi-axis flank milling would also be performed in future works.

In addition, other techniques of optimization, e.g. heuristic method or genetic algorithm, etc, should be examined. The objective is to reduce the computation burden in solving the OGA optimization problem and to produce more optimal control gains for the contour error reduction.

The gain modification pattern depends on all axis trajectories. This dependency has to be deeper analyzed to reduce the computation time. Ideally, it is expected to analyze the obtained gain variation profiles to eliminate or reduce the optimization stage. Moreover, using optimization results of similar trajectories is to produce gain modification profiles in function of trajectories.

Last but not least, the experimental tests of OGA are expected to be performed in the framework of an OPEN CNC.

Appendices

Appendix A

Nonlinear axis model

1 Motor model

Fig. A.1 shows a circuit model of DC motor, which is represented by the resistance R (Ohm), the inductance L (H), the back EMF (Electromotive Force) voltage E (V) and the voltage source U (V).

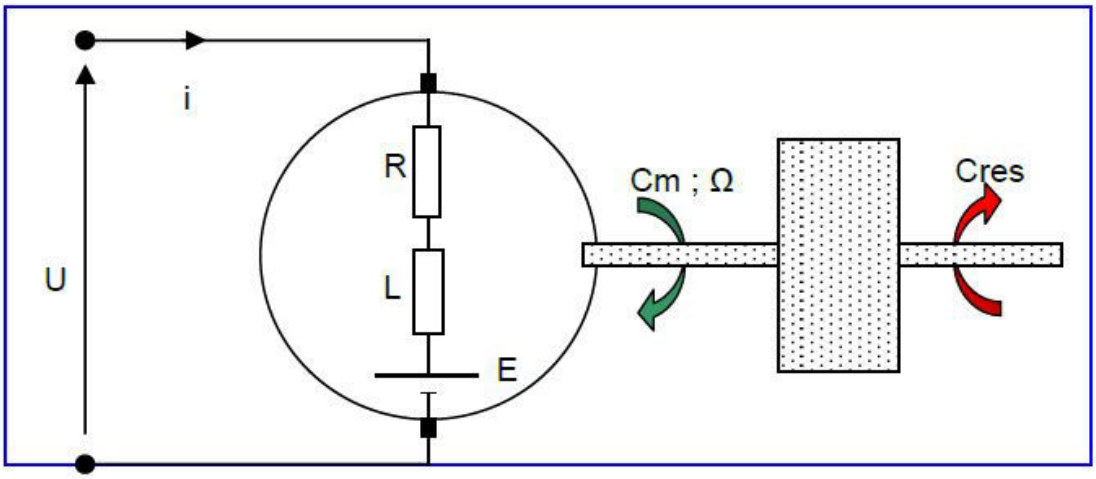

Figure A.1: Motor model [Prévost 2011b]

The back EMF voltage $E(t)$ is proportional to the angular velocity $\Omega(t)$ (rad/s) of the rotor in the motor, expressed as (A.1),

$$E(t) = K_e \Omega(t) \tag{A.1}$$

where K_e (V/rad/s) is the back EMF constant.

According to the Kirchhoff's voltage law, the electrical equation of the DC motor is described

by (A.2),

$$U(t) = Ri(t) + L\frac{di\,(t)}{dt} + E(t) \tag{A.2}$$

where $i(t)$ (A) is the armature current. This current passing through the rotor windings in the magnetic region of the stator generates the Laplace force. As a result, this force produces a torque C_m (N.m), leading to the rotation of the rotor, given as (A.3),

$$C_m\,(t) = K_t i(t) \tag{A.3}$$

where K_t (N.m/A) is the torque constant. Such a torque drives the external torque through the motor shaft, expressed by the dynamic equation in (A.4),

$$J\frac{d\Omega(t)}{dt} = C_m(t) - C_{res}\,(t) \tag{A.4}$$

where $C_{res}\,(t)$ (N.m) represents the resistant torque, which is due to the viscous and dry friction or the gravity of the Z axis; J (kg.m^2) the total equivalent rotor moment of inertia, that is obtained by (A.5),

$$J = J_{eq} = J_{mot} + J_{red} + \frac{J_{screw}}{N^2} + \frac{J_{axis}}{N^2} \tag{A.5}$$

where J_{mot}, J_{red}, J_{screw} and J_{axis} are the equivalent inertia of the motor, the gear reducer, the ball screw and the driven axis; N is the reduction factor.

Considering all of the equations from (A.1) to (A.5) in the Laplace domain s, the electric, eletro-mechanic and mechanic equations are obtained in (A.6), (A.7) and (A.8), respectively.

$$U(s) = R.I(s) + s.L.I(s) + K_e.\Omega(s) \tag{A.6}$$

$$C_m(s) = K_t.I(s) \tag{A.7}$$

$$C_m(s) = s.J.\Omega(s) + C_{res}(s) \tag{A.8}$$

The equations from (A.6) to (A.8) can be illustrated by the block diagram in Fig. A.2.

The above model is used for a rotary motor. For the linear motor, the detailed description can be found in [Prévost 2011b].

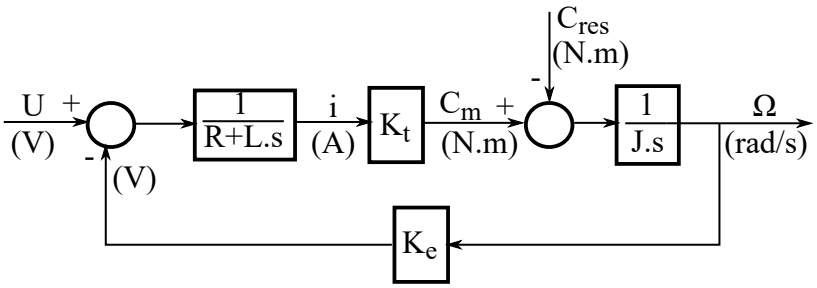

Figure A.2: Block diagram of motor model

2 Friction model

The friction law in each axis drive has been studied and proposed in [Prévost 2011a]. It is a combination of both viscous and Coulomb frictions.

The obtained friction model is characterized by (A.9),

$$\begin{cases} i_{fr} = ae^{bV} + ce^{dV} \quad if\ V \geq 0 \\ i_{fr} = -ae^{bV} + ce^{-dV} \quad if\ V \leq 0 \\ i_{fr} \in [-i_0; i_0] \quad if\ V = 0 \end{cases} \tag{A.9}$$

where i_{fr} (A) is the current caused by the friction, V (m/min) is the axis velocity, the parameters a, b, c, d and the value of the static friction current i_0 (A) are given in Table A.1.

Table A.1: Parameter of the friction model of Mikron machine [Prévost 2011a]

Friction parameter	Unit	Linear axes			Rotary axes	
		X axis	Y axis	Z axis	A axis	C axis
a	A.min/m	1.576	1.253	1.420	2.079	3.832
b		0.01965	0.01895	0.01650	0.02919	0.03248
c	A.min/m	-0.5332	-0.3629	-0.6301	-0.9891	-2.6177
d		-0.2801	-0.4026	-0.2625	-0.1946	-0.3441
i_0	A	1.043	0.890	0.790	1.090	1.216

Friction law and experimental results for X axis can be seen in Fig. A.3.

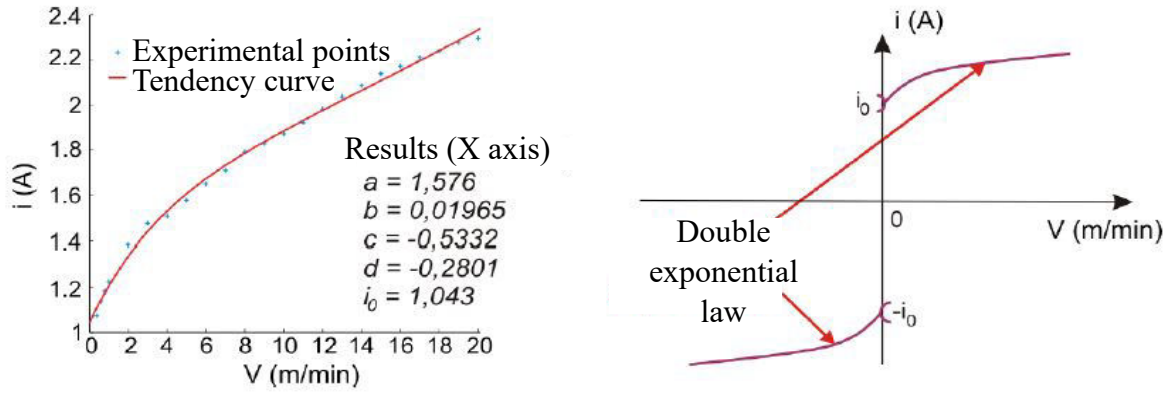

Figure A.3: Friction law and experimental results for X axis [Prévost 2011a]

In the concerned motor model in Fig. A.2, the resistant torque C_{res} derived from the impact of the friction model (A.9). Thus, the mechanic equation in (A.8) is rewritten as (A.10).

$$C_m(s) = s.J.\Omega(s) + K_t I_{fr}(s) \tag{A.10}$$

The motor model with the friction model (A.9) is illustrated in Fig. A.4,

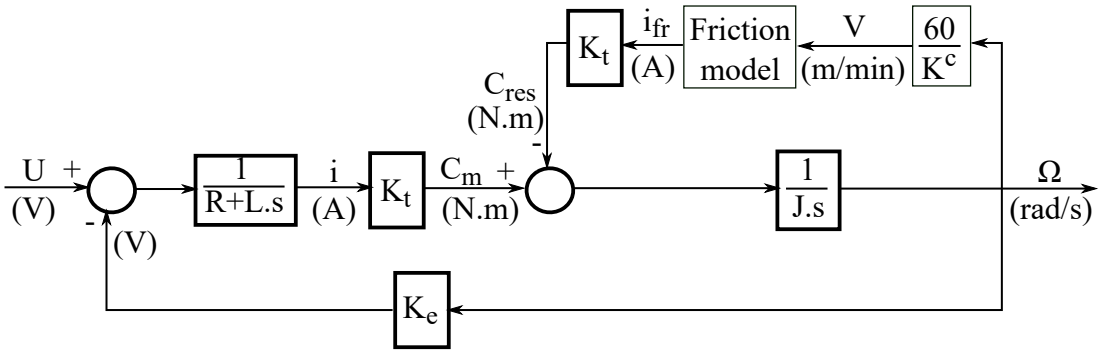

Figure A.4: Block diagram of motor model with friction model

in which K^c is the conversion factor from m to rad.

Appendix B

CAM configuration for tool path generation

Table B.1: CAM configuration for the programmed tool path generation

Parameters	Value/Configuration	Unit
Tool	Hemispheric tool	-
Type	Point milling	-
Tool radius	5	mm
Tool length	100	mm
Tilt angle	5	deg
Yaw angle	0	deg
Machining tolerance	10^{-4}	mm

Bibliography

[Abdel-Malek 1997] K. Abdel-Malek and H.J. Yeh. *Geometric representation of the swept volume using Jacobian rank-deficiency conditions*. Computer-Aided Design, vol. 29, no. 6, pages 457 – 468, 1997. (Cited on page 34.)

[Altintas 1994] Y. Altintas and W.K. Munasinghe. *A hierarchical open-architecture CNC system for machine tools*. CIRP Annals - Manufacturing Technology, vol. 43, no. 1, pages 349 – 354, 1994. (Cited on pages xx and 6.)

[Altintas 2000a] Y. Altintas. Manufacturing automation: Metal cutting mechanics, machine tool calibrations, and cnc design. Cambridge University Press, Cambridge, 2000. (Cited on pages xix, xx, 5 and 44.)

[Altintas 2000b] Y. Altintas, K. Erkorkmaz and W.H. Zhu. *Sliding Mode Controller Design for High Speed Feed Drives*. CIRP Annals - Manufacturing Technology, vol. 49, no. 1, pages 265–270, 2000. (Cited on pages xx, 5, 6, 47 and 49.)

[Altintas 2010] Y. Altintas and B. Sencer. *High speed contouring control strategy for five-axis machine tools*. CIRP Annals - Manufacturing Technology, vol. 59, no. 1, pages 417 – 420, 2010. (Cited on pages 22, 23, 24 and 25.)

[Altintas 2012] Y. Altintas and M.R. Khoshdarregi. *Contour error control of CNC machine tools with vibration avoidance*. CIRP Annals - Manufacturing Technology, vol. 61, no. 1, pages 335 – 338, 2012. (Cited on pages 23, 24 and 25.)

[APT 1983] APT. *Part programming manual format APT 360*. page 1, 1983. (Cited on page 13.)

[Barre 2005] P.J. Barre, R. Béarée, P. Borne and E. Dumetz. *Influence of a Jerk Controlled Movement Law on the Vibratory Behaviour of High-Dynamics Systems*. Journal of Intelligent and Robotic Systems, vol. 42, no. 3, pages 275–293, 2005. (Cited on pages xix, 5 and 38.)

[Bedi 2003] S. Bedi, S. Mann and C. Menzel. *Flank milling with flat end milling cutters*. Computer-Aided Design, vol. 35, no. 3, pages 293 – 300, 2003. (Cited on page 34.)

[Besset 2017] P. Besset and R. Béarée. *FI.filter-based online jerk-constrained trajectory generation.* Control Engineering Practice, vol. 66, pages 169 – 180, 2017. (Cited on page 13.)

[Beudaert 2012] X. Beudaert, S. Lavernhe and C. Tournier. *Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path.* International Journal of Machine Tools and Manufacture, vol. 57, pages 73–82, 2012. (Cited on pages xix, 5, 38 and 78.)

[Beudaert 2013] X. Beudaert. *Commande numérique ouverte : interpolation optimisée pour l'usinage 5 axes grande vitesse des surfaces complexes.* PhD thesis, ENS de Cachan, 2013. (Cited on pages x, xxi, 7, 13 and 40.)

[Beudaert 2014] X. Beudaert, S. Lavernhe and C. Tournier. *Direct trajectory interpolation on the surface using an open CNC.* The International Journal of Advanced Manufacturing Technology, vol. 75, no. 1, pages 535–546, 2014. (Cited on pages xx and 6.)

[Blackmore 1994] D. Blackmore, M.C. Leu and F. Shih. *Analysis and modelling of deformed swept volumes.* Computer-Aided Design. Special Issue: Mathematical methods for CAD, vol. 26, no. 4, pages 315 – 326, 1994. (Cited on page 34.)

[Bobrow 1985] J.E. Bobrow, S. Dubowsky and J.S. Gibson. *Time-Optimal Control of Robotic Manipulators Along Specified Paths.* The International Journal of Robotics Research, vol. 4, no. 3, pages 3–17, 1985. (Cited on page 37.)

[Bohez 2002] E.L.J. Bohez. *Compensating for systematic errors in 5-axis NC machining.* Computer-Aided Design, vol. 34, no. 5, pages 391 – 403, 2002. (Cited on pages ix and 19.)

[Cheng 2009] M.Y. Cheng and K.H. Su. *Contouring accuracy improvement using a tangential contouring controller with a fuzzy logic-based feedrate regulator.* The International Journal of Advanced Manufacturing Technology, vol. 41, no. 1, pages 75–85, 2009. (Cited on pages 48 and 49.)

[Chiou 2004] J.C.J. Chiou. *Accurate tool position for five-axis ruled surface machining by swept envelope approach.* Computer-Aided Design, vol. 36, no. 10, pages 967 – 974, 2004. (Cited on page 34.)

[Chiu 2001] G.C. Chiu and M. Tomizuka. *Contouring control of machine tool feed drive systems: a task coordinate frame approach.* IEEE Transaction on Control System Technology, vol. 9, no. 1, page 130–139, 2001. (Cited on pages 23, 24, 25 and 48.)

[Clarke 1987] D.W. Clarke, C. Mohtadi and P.S. Tuffs. *Generalized predictive control – Part I. and II.* Automatica, vol. 23, no. 2, pages 137–160, 1987. (Cited on pages 136 and 137.)

[Clarke 1989] D.W. Clarke and C. Mohtadi. *Properties of generalized predictive control.* 25, vol. 6, pages 859–875, 1989. (Cited on page 136.)

[Dagiloke 1995] I.F. Dagiloke, A. Kaldos, S. Douglas and B. Mills. *High-speed machining: An approach to process analysis.* Journal of Materials Processing Technology, vol. 54, no. 1, pages 82 – 87, 1995. (Cited on pages ix, xviii and 4.)

[Davis 2015] T.A. Davis, Y.C. Shin and B. Yao. *Adaptive Robust Control of Circular Machining Contour Error Using Global Task Coordinate Frame.* ASME. J. Manuf. Sci. Eng., vol. 137, no. 1, page 8, 2015. (Cited on pages xx, 6, 48 and 49.)

[Dong 2005] J.J. Dong and J.A. Stori. *A Generalized Time-Optimal Bi-Directional Scan Algorithm for Constrained Feedrate Optimization.* ASME International Mechanical Engineering Congress and Exposition, Manufacturing Engineering and Materials Handling Engineering, vol. 128, no. 2, pages 379–390, 2005. (Cited on page 38.)

[Duc 1998] E. Duc. *Usinage des formes gauches, contribution à l'amélioration de la qualité des trajectoires d'usinage.* PhD thesis, ENS de Cachan, 1998. (Cited on page 15.)

[Duc 1999] E. Duc, C. Lartigue, C. Tournier and P. Bourdet. *A New Concept for the Design and the Manufacturing of Free-Form Surfaces: The Machining Surface.* CIRP Annals - Manufacturing Technology, vol. 48, no. 1, pages 103 – 106, 1999. (Cited on pages ix and 16.)

[Dumur 2008] D. Dumur, M. Susanu and M. Aubourg. *Complex form machining with axis drive predictive control.* CIRP Annals - Manufacturing Technology, vol. 57, no. 1, pages 399 – 402, 2008. (Cited on pages 47, 49 and 133.)

[El-Khalick 2011] M.A. El-Khalick and N. Uchiyama. *Discrete-time model predictive contouring control for biaxial feed drive systems and experimental verification.* Mechatronics, vol. 21, no. 6, pages 918–926, 2011. (Cited on pages 48 and 49.)

[Erkorkmaz 2001a] K. Erkorkmaz and Y. Altintas. *High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation*. International Journal of Machine Tools and Manufacture, vol. 41, no. 9, pages 1323–1345, 2001. (Cited on pages xix, 5 and 38.)

[Erkorkmaz 2001b] K. Erkorkmaz and Y. Altintas. *High speed CNC system design. Part III: high speed tracking and contouring control of feed drives*. International Journal of Machine Tools and Manufacture, vol. 41, no. 11, pages 1637 – 1658, 2001. (Cited on pages xx, 5, 6, 22, 47 and 49.)

[Erkorkmaz 2006] K. Erkorkmaz, C.H. Yeung and Y. Altintas. *Virtual CNC system. Part II. High speed contouring application*. International Journal of Machine Tools and Manufacture, vol. 46, no. 10, pages 1124 – 1138, 2006. (Cited on pages 23, 24, 25, 29 and 49.)

[Erkorkmaz 2017] K. Erkorkmaz, Q.G. Chen, M.Y. Zhao, X. Beudaert and X.S. Gao. *Linear programming and windowing based feedrate optimization for spline toolpaths*. CIRP Annals, vol. 66, no. 1, pages 393 – 396, 2017. (Cited on page 38.)

[Ghaffari 2016] A. Ghaffari and A.G. Ulsoy. *Dynamic Contour Error Estimation and Feedback Modification for High-Precision Contouring*. IEEE/ASME Transactions on Mechatronics, vol. 21, no. 3, pages 1732–1741, 2016. (Cited on pages 23, 24 and 25.)

[Huo 2012] F. Huo and A.N. Poo. *Improving contouring accuracy by using generalized cross-coupled control*. International Journal of Machine Tools and Manufacture, vol. 63, pages 49–57, 2012. (Cited on pages 48 and 49.)

[Jung 2002] Y.H. Jung, D.W. Lee, J.S. Kim and H.S. Mok. *NC post-processor for 5-axis milling machine of table-rotating/tilting type*. Journal of Materials Processing Technology, vol. 130, pages 641 – 646, 2002. AFDM 2002 S.I. (Cited on page 13.)

[Kamalzadeh 2007] A. Kamalzadeh and K. Erkorkmaz. *Accurate tracking controller design for high-speed drives*. International Journal of Machine Tools and Manufacture, vol. 47, no. 9, pages 1393 – 1400, 2007. Selected papers from the 2nd International Conference on High Performance Cutting. (Cited on pages 47 and 49.)

[Khoshdarregi 2014] M.R. Khoshdarregi, S. Tappe and Y. Altintas. *Integrated Five-Axis Trajectory Shaping and Contour Error Compensation for High-Speed CNC Machine Tools*.

IEEE/ASME Transactions on Mechatronics, vol. 19, no. 6, pages 1859–1871, 2014. (Cited on pages xx, 5, 23, 24, 25, 48 and 49.)

[Kim 1995] K.I. Kim and K. Kim. *A new machine strategy for sculptured surfaces using offset surface*. International Journal of Production Research, vol. 33, no. 6, pages 1683–1697, 1995. (Cited on pages ix and 17.)

[Koren 1980] Y. Koren. *Cross-Coupled Biaxial Computer Control for Manufacturing Systems*. Journal of Dynamic Systems, Measurement, and Control, vol. 102, no. 4, pages 265–272, 1980. (Cited on pages xx, 6, 23, 24, 25, 48 and 49.)

[Koren 1983] Y. Koren. *Computer Control of Manufacturing Systems*. McGraw-Hill, 1983. (Cited on page 22.)

[Koren 1991] Y. Koren and C.C. Lo. *Variable-Gain Cross-Coupling Controller for Contouring*. CIRP Annals - Manufacturing Technology, vol. 40, no. 1, pages 371–374, 1991. (Cited on pages xx, 6, 47, 48 and 49.)

[Kovács 2012] E. Kovács. *Rotation about an arbitrary axis and reflection through an arbitrary plane*. Annales Mathematicae et Informaticae, vol. 40, pages 175–186, 2012. (Cited on page 28.)

[Landau 1990] I.D. Landau. System identification and control design: Using p.i.m. + software. Academic press, 1990. (Cited on page 136.)

[Langeron 2004] J.M. Langeron, E. Duc, C. Lartigue and P. Bourdet. *A new format for 5-axis tool path computation, using Bspline curves*. Computer-Aided Design, vol. 36, no. 12, pages 1219 – 1229, 2004. (Cited on page 18.)

[Lauwers 2003] B. Lauwers, P. Dejonghe and J.P. Kruth. *Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation*. Computer-Aided Design, vol. 35, no. 5, pages 421 – 432, 2003. (Cited on page 16.)

[Lavernhe 2006] S. Lavernhe. *Prise en compte des contraintes associées au couple MO-CN en génération de trajectoires 5 axes UGV*. PhD thesis, ENS de Cachan, 2006. (Cited on pages ix, 18 and 26.)

[Li 2016] X. Li, H. Zhao, X. Zhao and H. Ding. *Dual sliding mode contouring control with high accuracy contour error estimation for five-axis CNC machine tools.* International Journal of Machine Tools and Manufacture, vol. 108, no. Supplement C, pages 74 – 82, 2016. (Cited on pages 23, 24 and 25.)

[Lin 1996] R. Lin and Y.Y. Koren. *Efficient Tool-Path Planning for Machining Free-Form Surfaces.* ASME. Journal of Engineering for Industry, vol. 118, no. 1, pages 20–28, 1996. (Cited on pages ix and 18.)

[Lin 2006] F.J. Lin, H.J. Shieh, P.H. Shieh and P.H. Shen. *An adaptive recurrent-neural-network motion controller for X-Y table in CNC Machine.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 36, no. 2, pages 286–299, 2006. (Cited on page 6.)

[Mori 2013] M. Mori and M. Fujishima. *Remote Monitoring and Maintenance System for CNC Machine Tools.* Procedia CIRP, vol. 12, pages 7 – 12, 2013. Eighth CIRP Conference on Intelligent Computation in Manufacturing Engineering. (Cited on page 13.)

[Neugebauer 2007] R. Neugebauer, B. Denkena and K. Wegener. *Mechatronic Systems for Machine Tools.* CIRP Annals - Manufacturing Technology, vol. 56, no. 2, pages 657 – 686, 2007. (Cited on page 13.)

[Ngo 2013] P.D. Ngo and Y.C. Shin. *Milling contour error control using multilevel fuzzy controller.* The International Journal of Advanced Manufacturing Technology, vol. 66, no. 9-12, pages 1641—1655, 2013. (Cited on page 6.)

[Omirou 2005] S.L. Omirou and A.K. Barouni. *Integration of new programming capabilities into a CNC milling system.* Robotics and Computer-Integrated Manufacturing, vol. 21, no. 6, pages 518 – 527, 2005. (Cited on page 13.)

[Pechard 2011] P.Y. Pechard. *Génération de trajectoire d'usinage grand vitesse 5 axes par flanc d'outil.* PhD thesis, ENS de Cachan, 2011. (Cited on pages x, 34 and 45.)

[Prévost 2011a] D. Prévost. *Modélisation avancée du couple Machine Outil / Commande Numérique dans un contexte UGV pour l'optimisation du calcul de trajectoires multi axes.* PhD thesis, ENS de Cachan, 2011. (Cited on pages x, xiii, xiv, xxi, 7, 24, 30, 45, 46, 54, 84, 165 and 166.)

[Prévost 2011b] D. Prévost, S. Lavernhe, C. Lartigue and D. Dumur. *Feed drive modelling for the simulation of tool path tracking in multi-axis High Speed Machining.* International Journal of Mechatronics and Manufacturing Systems, vol. 4, no. 3-4, pages 266–284, 2011. (Cited on pages xiii, 163 and 164.)

[Pritschow 1996] G. Pritschow. *On the Influence of the Velocity Gain Factor on the Path Deviation.* CIRP Annals - Manufacturing Technology, vol. 45, no. 1, pages 367 – 371, 1996. (Cited on page 52.)

[Renton 2000] D. Renton and M.A. Elbestawi. *High speed servo control of multi-axis machine tools.* International Journal of Machine Tools and Manufacture, vol. 40, no. 4, pages 539 – 559, 2000. (Cited on page 38.)

[Rodriguez-Ayerbe 2005] P. Rodriguez-Ayerbe and D. Dumur. *Generalized Predictive Control robustification under frequency and time-domain constraints.* IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pages 577–587, 2005. (Cited on page 142.)

[Rodriguez-Ayerbe 2014] P. Rodriguez-Ayerbe, D. Dumur and S. Lavernhe. *Axis control using model predictive control: identification and friction effect reduction.* 3rd Int Conf on Virtual Machi Process Tech, page 8, 2014. (Cited on pages 47, 49, 133, 134 and 142.)

[Schmitz 2001] T. Schmitz, M. Davies, B. Dutterer and J. Ziegert. *The application of high-speed CNC machining to prototype production.* International Journal of Machine Tools and Manufacture, vol. 41, no. 8, pages 1209 – 1228, 2001. (Cited on page 13.)

[Seimens 2005] Seimens. *Siemens Configuration Manual Simodrive 611 / Masterdrives MC, 1FT6 Synchronous Motors.* pages 144 – 145, 2005. (Cited on pages 62 and 87.)

[Sencer 2008] B. Sencer, Y. Altintas and E. Croft. *Feed optimization for five-axis CNC machine tools with drive constraints.* International Journal of Machine Tools and Manufacture, vol. 48, no. 7, pages 733 – 745, 2008. (Cited on page 38.)

[She 2007] C.H. She and C.C. Chang. *Design of a generic five-axis postprocessor based on generalized kinematics model of machine tool.* International Journal of Machine Tools and Manufacture, vol. 47, no. 3, pages 537 – 545, 2007. (Cited on page 13.)

[Shiller 1990] Z. Shiller and H.H. Lu. *Robust computation of path constrained time optimal motions*. In Proceedings., IEEE International Conference on Robotics and Automation, pages 144–149 vol.1, 1990. (Cited on page 38.)

[Shin 1985] K. Shin and N. McKay. *Minimum-time control of robotic manipulators with geometric path constraints*. IEEE Transactions on Automatic Control, vol. 30, no. 6, pages 531–541, 1985. (Cited on page 38.)

[Susanu 2005] M. Susanu. *Commande prédictive hiérarchisée pour le positionnement des axes de machine-outil à architecture ouverte*. PhD thesis, Univ. Paris Sud, 2005. (Cited on pages x, xxi, 7, 44 and 45.)

[Susanu 2006] M. Susanu and D. Dumur. *Hierarchical Predictive Control within an Open Architecture Virtual Machine Tool*. CIRP Annals - Manufacturing Technology, vol. 55, no. 1, pages 389–392, 2006. (Cited on pages 6 and 133.)

[Tang 2012] L. Tang and R. G. Landers. *Predictive Contour Control With Adaptive Feed Rate*. IEEE/ASME Transactions on Mechatronics, vol. 17, no. 4, pages 669–679, 2012. (Cited on page 6.)

[Tang 2013] L. Tang and R.G. Landers. *Multiaxis Contour Control - the State of the Art*. IEEE Transactions on Control Systems Technology, vol. 21, no. 6, pages 1997–2010, 2013. (Cited on pages xx, 5 and 47.)

[Timar 2005] S.D. Timar, R.T. Farouki, T.S. Smith and C.L. Boyadjieff. *Algorithms for time–optimal control of CNC machines along curved tool paths*. Robotics and Computer-Integrated Manufacturing, vol. 21, no. 1, pages 37 – 53, 2005. (Cited on page 38.)

[Yang 2005] J. Yang and K. Abdel-Malek. *Approximate swept volumes of NURBS surfaces or solids*. Computer Aided Geometric Design, vol. 22, no. 1, pages 1 – 26, 2005. (Cited on page 34.)

[Yang 2015a] J. Yang and Y. Altintas. *A generalized on-line estimation and control of five-axis contouring errors of CNC machine tools*. International Journal of Machine Tools and Manufacture, vol. 88, pages 9–23, 2015. (Cited on pages 22, 23, 24 and 25.)

[Yang 2015b] S. Yang, A.H. Ghasemi, X. Lu and C.E. Okwudire. *Precompensation of servo contour errors using a model predictive control framework*. International Journal of Machine Tools and Manufacture, vol. 98, pages 50–60, 2015. (Cited on pages 23, 24, 25, 48 and 49.)

[Yao 1997] B. Yao. *High performance adaptive robust control of nonlinear systems: a general framework and new schemes*. Decision and Control, Proceeding of the 36th IEEE Conference, vol. 3, pages 2489–2494, 1997. (Cited on pages xx and 6.)

[Yao 2012] B. Yao, C. Hu and Q. Wang. *An Orthogonal Global Task Coordinate Frame for Contouring Control of Biaxial Systems*. IEEE/ASME Transactions on Mechatronics, vol. 17, no. 4, pages 622–634, 2012. (Cited on pages 22, 23, 24, 25, 48 and 49.)

[Younkin 2003] G.W. Younkin. Industrial servo control systems: Fundamentals and applications. Marcel Dekker, Inc, 2003. (Cited on page 60.)

[Yuen 2013] A. Yuen, K. Zhang and Y. Altintas. *Smooth trajectory generation for five-axis machine tools*. International Journal of Machine Tools and Manufacture, vol. 71, pages 11 – 19, 2013. (Cited on page 13.)

[Zhang 2013] K. Zhang, A. Yuen and Y. Altintas. *Pre-compensation of contour errors in five-axis CNC machine tools*. International Journal of Machine Tools and Manufacture, vol. 74, pages 1–11, 2013. (Cited on pages 23, 24, 25, 48 and 49.)

[Zhang 2015] D. Zhang, J. Yang, Yu. Chen and Yo. Chen. *A two-layered cross coupling control scheme for a three-dimensional motion control system*. International Journal of Machine Tools and Manufacture, vol. 98, pages 12–20, 2015. (Cited on pages 48 and 49.)

[Zhao 2013] G. Zhao, H. An and Q. Zhao. *Contour Error Coupled-Control Strategy based on Line Interpolation and Curve Interpolation*. Journal of Computers, vol. 8, no. 6, pages 1512–1519, 2013. (Cited on pages xx and 6.)

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Commande à gains variables de l'erreur de contour pour l'usinage multiaxes

Mots clefs : lois de commande adaptatives; erreur de contour; usinage grande vitesse

Résumé : Les techniques d'usinage avancées sont un élément indispensable du développement des industries manufacturières. L'une de ces techniques, l'usinage à grande vitesse, constitue le sujet principal de cette thèse de doctorat. Ainsi, l'objectif majeur des travaux vise à améliorer la précision de contour dans le contexte de l'usinage multiaxes à grande vitesse de surfaces de forme libre, en agissant directement au niveau des boucles de commande d'axe. Pour cela, une première étape consiste à élaborer une stratégie permettant d'estimer le plus précisément possible l'erreur de contour pour différentes configurations de l'outil. Cette erreur de contour est ensuite minimisée grâce à l'adaptation hors ligne, pour un profil de pièce donné, des gains proportionnel et d'anticipation des régulateurs des boucles d'asservissement de la position de chaque axe. L'adaptation de ces gains est réalisée via un algorithme d'optimisation à l'aide d'un modèle non-linéaire du comportement de la machine, en considérant en particulier les frottements sur chacun des axes. L'optimisation permettant d'obtenir les gains des correcteurs des boucles de régulation tient compte des contraintes en termes de limitations cinématiques des axes (vitesse, accélération et jerk), de stabilité des boucles d'asservissement et de limites au niveau des courants des moteurs. Afin d'en faciliter la mise en œuvre dans un cadre industriel, les stratégies développées s'avèrent directement implantables au sein des commandes numériques actuellement sur le marché, exploitant toutes les possibilités de la structure de commande classique de l'entraînement d'axe.

Title : Variable gain contouring control for multi-axis machine tools

Keywords : adaptive control laws; contour error; high speed machining

Abstract : The advanced machining techniques are always the backbone of the manufacturing industries. Among such techniques, high speed machining is the main subject of this PhD thesis. Indeed, the main objective of this work is to improve the contouring accuracy in multi-axis high speed machining of free-form surfaces, directly acting inside the axis control loops. To do that, a first step aims at elaborating a strategy to estimate as accurately as possible the contour error for different tool configurations. This contour error is then minimized by means of an off-line adaptation for a given profile of the proportional and feedforward gains of the axis position loop controllers. This gain adaptation is performed via an optimization algorithm that considers a nonlinear model of the machine behaviour, in particular including friction related to each axis. This optimization leading to the controllers gains takes into account several constraints, including the axis kinematic (velocity, acceleration and jerk) limitations, the stability of the controlled loops and the motor current limits. Finally, to help their integration within an industrial framework, the developed strategies can be directly implemented in commercial CNC, by exploiting all possibilities of the classical control structure of axis drive.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France