



HAL
open science

Contribution à l'analyse de performance des robots parallèles

Ranjan Jha

► **To cite this version:**

Ranjan Jha. Contribution à l'analyse de performance des robots parallèles. Robotique [cs.RO]. Ecole Centrale de Nantes (ECN), 2016. Français. NNT: . tel-01524856

HAL Id: tel-01524856

<https://hal.science/tel-01524856v1>

Submitted on 19 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse de Doctorat

Ranjan JHA

*Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Ecole Centrale de Nantes
sous le label de L'Université Nantes Angers Le Mans*

École doctorale : STIM

Discipline : Sciences pour l'ingénieur
Unité de recherche : IRCCyN

Soutenue le 7 juillet 2016

Contributions to the performance analysis of parallel robots

JURY

Président : **Said ZEGHLOUL**, Professeur, Université de Poitiers

Rapporteurs : **Jean-Pierre MERLET**, Directeur de Recherche, INRIA
Grigore GOGU, Professeur, IFMA

Examineurs : **Damien CHABLAT**, Directeur de Recherche, CNRS
Guillaume MOROZ, Chargé de Recherche, INRIA
Fabrice ROUILLIER, Directeur de Recherche, INRIA

Invité(s) :

Directeur de Thèse : **Damien CHABLAT**, Directeur de Recherche, CNRS

Co-directeur de Thèse : **Fabrice ROUILLIER**, Directeur de Recherche, INRIA, **Guillaume MOROZ**, Chargé de Recherche, INRIA

Thèse de Doctorat

Ranjan JHA

Contributions to the performance analysis of parallel robots

Résumé

Cette thèse apporte des contributions pour plusieurs problèmes associés à la planification de mouvements pour les robots parallèles. Ces différents problèmes sont classés en quatre catégories: l'analyse de l'espace de travail et de l'espace articulaire ; les domaines d'unicité ; la planification des trajectoires et l'analyse de précision. L'analyse de l'espace de travail et de l'espace articulaire différencient les régions possédant un nombre constant de solutions aux modèles géométrique inverses et des directs en utilisant une décomposition algébrique cylindrique. Une méthode d'élimination utilisant des bases de Gröbner est utilisée pour calculer les singularités parallèles et sérielles des manipulateurs parallèles étudiés. Les surfaces caractéristiques sont également calculées pour définir les domaines d'unicité dans l'espace de travail. Une extension de la notion d'aspect est proposée pour les robots parallèles possédant plusieurs modes de fonctionnement, d'assemblage et d'opération. Une méthode algébrique est proposée pour vérifier la faisabilité de toutes les trajectoires données dans l'espace de travail pour régler le problème bien connu qui se pose lorsqu'il existe une configuration singulière entre les deux postures de la plate-forme mobile lorsque l'on discrétise la trajectoire avec une approche classique. Appliquée au robot NaVARo, un robot parallèle possédant plusieurs modes d'actionnement, une interface Matlab est présentée pour générer sa prise d'origine et ses mouvements en utilisant uniquement le modèle géométrique inverse. Pour l'Orthoglide 5-axes, une analyse de précision est réalisée pour estimer l'erreur de position de l'outil associée aux erreurs produites par la boucle de régulation PID. Le modèle d'erreur proposé, qui est basé sur les propriétés statiques et dynamiques de l'Orthoglide, permet d'estimer les erreurs de positionnement dans l'espace cartésien.

Mots clés

Manipulateur parallèles, planification des trajectoires, décomposition cylindrique algébrique, singularités, base de Gröbner, paramètres de conception, les indices de performance, modes d'opération

Abstract

This doctoral thesis focuses on the different aspects which are associated with efficient planning of desired tasks for parallel robots. These different aspects are mainly categorized in four parts, namely: workspace and joint space analysis, uniqueness domains, trajectory planning and accuracy analysis. The workspace and joint space analysis differentiate the regions with different number of inverse kinematic solutions and direct kinematic solutions using a cylindrical algebraic decomposition algorithm, respectively. The influence of design parameters and joint limits on the workspace boundaries for the parallel robots are reported. Gröbner based elimination methods are used to compute the parallel and serial singularities of the manipulator under study. The descriptive analysis of a family of delta like robots is presented by using algebraic tools to induce the an estimation about the complexity in representing the singularities in the workspace and the joint space. The generalized notions of aspects and uniqueness domains are defined for the parallel robot with several operation modes. The characteristic surfaces are also computed to define the uniqueness domains in the workspace. An algebraic method is proposed to check the feasibility of any given trajectory in the workspace to address the well known problem which arises when there exists a singular configuration between the two poses of the end-effectors while discretizing the path with a classical approach. A Framework for the control loop of a parallel robot with several actuation modes is presented, which uses only the inverse geometric model. The accuracy analysis focuses on the estimation of errors in the pose of the end effector due to the joint's errors produced by the PID control loop. The proposed error model, which is based on the static and dynamic properties of the Orthoglide, helps in estimating the error in the Cartesian workspace

Key Words

Parallel manipulators, trajectory planning, cylindrical algebraic decomposition, singularities, Gröbner basis, design parameters, performance indices, operation modes

Acknowledgements

It is very well said that *Solving a problem for which you know there's an answer is like climbing a mountain with a guide, along a trail someone else has laid*. I would have never been able to complete my dissertation without the guidance, help and support of the kind people around me, to only some of whom it is possible for me to give a particular mention here.

At every step God helped me not to waste the amazing opportunities that came before me and gave me confidence to utilize them. First of all, I would like to thank the almighty God for establishing me to complete this thesis. I would like to thank my parents, sister, brother-in-law and other family members for their support and great patience all the time. They have given me their unequivocal support throughout, as always, for which my mere expression of thanks likewise does not suffice.

My thesis at IRCCyN would not have been possible without the support, guidance and suggestions of many people. I would like to thank my thesis supervisors Prof. D. Chablat, Prof. F. Rouillier and Dr. G. Moroz, for giving me an opportunity to work with them.

I am deeply indebted to my research supervisor, Professor D. Chablat for presenting me such an interesting thesis topic. Each meeting with him added invaluable aspects to the implementation and broadened my perspective. He has guided me with his invaluable suggestions, lightened up the way in my darkest times and encouraged me a lot in the academic life. From him I have learned to think critically, to select problems, to solve them and to present their solutions.

I would like to address special thanks to the jury members, Prof. S. Zegloul, Prof. J.P Merlet and Prof. G. Gogu for their valuable comments which helped to considerably improve the quality of the thesis.

I also thank robotics team, International Office members, my colleagues and lab mates at IRCCyN, Ecole Centrale de Nantes for being a constant source of motivation, enthusiasm and encouragement and making it a fun place to work at. I would like to extend my sincerest thanks to the European Commission for their generous fellowship (Erasmus Mundus INDIA4EU II) which allowed me to work on my thesis.

RANJAN JHA

Résumé

Cette thèse apporte des contributions pour plusieurs problèmes associés à la planification de mouvements pour les robots parallèles. Ces différents problèmes sont classés en quatre catégories: l'analyse de l'espace de travail et de l'espace articulaire ; les domaines d'unicité ; la planification des trajectoires et l'analyse de précision. L'analyse de l'espace de travail et de l'espace articulaire différencient les régions possédant un nombre constant de solutions aux modèles géométrique inverses et directs en utilisant une décomposition algébrique cylindrique. Une méthode d'élimination utilisant des bases de Gröbner est utilisée pour calculer les singularités parallèles et sérielles des manipulateurs parallèles étudiés. Les surfaces caractéristiques sont également calculées pour définir les domaines d'unicité dans l'espace de travail. Une extension de la notion d'aspect est proposée pour les robots parallèles possédant plusieurs modes de fonctionnement, d'assemblage et d'opération. Une méthode algébrique est proposée pour vérifier la faisabilité de toutes les trajectoires données dans l'espace de travail a fin de régler le problème bien connu qui se pose lorsqu'il existe une configuration singulière entre les deux postures de la plate-forme mobile lorsque l'on discrétise la trajectoire avec une approche classique. Appliquée au robot NaVARo, un robot parallèle possédant plusieurs modes d'actionnement, une interface Matlab est présentée pour générer sa prise d'origine et ses mouvements en utilisant uniquement le modèle géométrique inverse. Pour l'Orthoglide 5-axes, une analyse de précision est réalisée pour estimer l'erreur de position de l'outil associée aux erreurs produites par la boucle d'asservissement (PID). Le modèle d'erreur proposé, qui est basé sur les propriétés statiques et dynamiques de l'Orthoglide, permet d'estimer les erreurs de positionnement dans l'espace cartésien.

Mots-clés: Manipulateur parallèles, planification des trajectoires, décomposition cylindrique algébrique, singularités, base de Gröbner, paramètres de conception, modes d'opération.

Abstract

This doctoral thesis focuses on the different aspects which are associated with efficient planning of desired tasks for parallel robots. These different aspects are mainly categorized in four parts, namely: workspace and joint space analysis, uniqueness domains, trajectory planning and accuracy analysis. The workspace and joint space analysis differentiate the regions with different number of inverse kinematic solutions and direct kinematic solutions using a cylindrical algebraic decomposition algorithm, respectively. The influence of design parameters and joint limits on the workspace boundaries for the parallel robots are reported. Gröbner based elimination methods are used to compute the parallel and serial singularities of the manipulator under study. The descriptive analysis of a family of delta like robots is presented by using algebraic tools to induce an estimation about the complexity in representing the singularities in the workspace and the joint space. The generalized notions of aspects and uniqueness domains are defined for the parallel robot with several operation modes. The characteristic surfaces are also computed to define the uniqueness domains in the workspace. An algebraic method is proposed to check the feasibility of any given trajectory in the workspace to address the well known problem which arises when there exists a singular configuration between the two poses of the end-effectors while discretizing the path with a classical approach. A Framework for the control loop of a parallel robot with several actuation modes is presented , which uses only the inverse geometric model. The accuracy analysis focuses on the estimation of errors in the pose of the end effector due to the joint's errors produced by the PID control loop. The proposed error model, which is based on the static and dynamic properties of the Orthoglide, helps in estimating the error in the Cartesian workspace.

Keywords: Parallel Robots, Trajectory Planning, Cylindrical Algebraic Decomposition, Singularities, Gröbner Basis, Design Parameters, Operation Modes, Workspace.

Contents

Acknowledgements	i
Résumé	iii
Abstract	v
Nomenclature	1
Introduction	3
I Definitions and Mathematical Tools	11
1.1 Parallel Robots	11
1.1.1 Create Manipulator with SIROPA	12
1.1.2 Workspace and Jointspace	12
1.2 Kinematics : IKP and DKP	15
1.3 Singularities	16
1.3.1 Parallel Singularities	17
1.3.2 Serial Singularities	18
1.4 Assembly modes and Working modes	18
1.5 Operation Modes	20
1.6 Aspects and Uniqueness Domains	21
1.6.1 Aspect for parallel manipulators with single IKS	21
1.6.2 Aspect for parallel manipulators with multiple IKS	21
1.6.3 Characteristic surfaces	21
1.6.4 Basic components and Basic regions	22
1.6.5 Uniqueness domains	24

1.7 Non-Singular assembly mode changing trajectories	25
1.8 Standard Bases (Gröebner Bases)	26
1.9 Discriminant Variety and Cylindrical Algebraic Decomposition	26
1.10 Conclusions	29
II Workspace and Joint Space Analysis	31
II.1 Introduction	31
II.2 Manipulators Under Study	32
II.2.1 Orthoglide Architecture and Kinematics	32
II.2.2 Hybridglide Architecture and Kinematics	33
II.2.3 Triaglide Architecture and Kinematics	33
II.2.4 UraneSX Architecture and Kinematics	34
II.2.5 Reconfigurable 3-RPS Parallel Robot	35
II.2.6 NaVARo 3RRR Planar Parallel Robot	38
II.3 Workspace Analysis of a Delta-like Family Robot	39
II.4 Joint space Analysis of a Delta-like Family Robot	40
II.5 Workspace Analysis of the 3RPS Parallel robot	41
II.5.1 Operation Modes	42
II.5.2 Cylindrical Algebraic Decomposition	43
II.5.3 Workspace: 3T Projection Space for OM^1 and OM^2	44
II.6 Joint space Analysis of the 3RPS Parallel robot	44
II.7 Conclusions	48
III Aspects and Uniqueness Domains	51
III.1 Introduction	51
III.2 Singularities: Delta-Like Family Robot	52
III.2.1 Parallel Singularities: Projection in workspace and jointspace	52
III.2.2 Serial Singularities: Projection in workspace and jointspace	54
III.2.3 Complexity in Singularities	56
III.3 Parallel Singularities of 3RPS Parallel robot	57
III.4 Influence of design parameter on Parallel Singularities	58
III.5 Aspect for an Operation mode	61
III.6 Characteristic surfaces for an operation mode	62
III.7 Uniqueness domains	62
III.8 Non Singular assembly mode Trajectories	67
III.9 Conclusions	69
IV Trajectory Planning	71
IV.1 Introduction	71
IV.2 Methodology	72
IV.3 Method Validation	75
IV.3.1 Manipulator Architecture and Kinematics	75
IV.3.2 Trajectory definition in the workspace	78
IV.3.3 Projection in the joint space	79
IV.3.4 Singularity analysis	81

IV.4 Parallel Manipulator with Several Actuation Modes	84
IV.4.1 Robot under study	86
IV.4.2 Transmission System	87
IV.4.3 Control Hardware and Software	88
IV.4.4 Sensor Placements	88
IV.4.5 Graphic user interface	89
IV.5 Homing	89
IV.5.1 Introduction	89
IV.5.2 Direct kinematics of the 3-RPR	89
IV.5.3 Direct kinematics of the 3-RRR	90
IV.5.4 Homing procedure	91
IV.6 Motions of the robot	92
IV.6.1 Inverse kinematic model	92
IV.6.2 Jacobian matrices	94
IV.6.3 Actuation mode changing	95
IV.6.4 Motions of the robot	95
IV.7 Conclusions	95
V Prospective Work: Evaluation of the Accuracy	97
V.1 Introduction	97
V.2 Architecture & Kinematics: Orthoglide 5-AXIS	98
V.2.1 Direct Kinematics Model:	99
V.2.2 Inverse Kinematics Model:	100
V.2.3 Parallel Singularities	101
V.3 Trajectory Planning	101
V.4 Error Analysis	105
V.5 Error Model	107
V.6 Projection of Joint Errors in the Workspace	109
V.7 Influence of Starting positions on the joints errors	111
V.7.1 Joints errors for different starting positions	111
V.7.2 Projection of Joint errors in the Cartesian space	116
V.8 Conclusions	118
VI Conclusions and Future Work	119
VI.1 Summary	119
VI.2 Main Contributions	119
VI.3 Future Work	122
Personal Publications	125
A Appendix Mathematical Expressions	127
A.1 Manipulator Creation	127
A.2 Parallel Singularities	128
A.3 Serial Singularities	130
A.4 Reconfigurable 3RPS Parallel Robot	130
B Appendix SIROPA Library	139
B.1 SIROPA	139
B.2 Algebraic Tools	146

C Appendix Timeline	149
Bibliography	151

List of Figures

I.1	Projection of parallel Singularity curves	17
I.2	Projection of serial Singularity curves	18
I.3	Assembly modes of orthoglide	19
I.4	Postures of orthoglide	19
I.5	Octree model of the characterstic surfaces	22
I.6	Octree model of the basic regions	23
I.7	Octree model of the basic components	23
I.8	Octree model of the uniqueness domain	24
I.9	Non-singular assembly-mode changing trajectory	25
II.1	Configuration plot for Orthoglide robot	33
II.2	Configuration plot for Hybridglide robot	33
II.3	Configuration plot for Triaglide robot	34
II.4	Configuration plot for UraneSX robot	34
II.5	3-RPS parallel robot	35
II.6	Virtual model of 3-RPS parallel manipulator with different design parameter k, k = 0.5 (a), k = 1 (b), k = 1.5 (c)	36
II.7	The NaVARo robot	38
II.8	The virtual model of the NaVARo robot	38
II.9	Workspace plot for delta robots	39
II.10	Jointspace plot for delta robots	41
II.11	Aspects for OM^1 with $\det(\mathbf{A}) < 0$ (a) and $\det(\mathbf{A}) > 0$ (b) and aspects for OM^2 with $\det(\mathbf{A}) < 0$ (c) and $\det(\mathbf{A}) > 0$ (d)	42

II.12 Slice of workspace for OM^1 with different design parameter and $p_z = 2$, $k = 0.5$ (a), $k = 1$ (b), $k = 2$ (c) and $k = 3$ (d) for OM^2 $k = 0.5$ (e), $k = 1$ (f), $k = 2$ (g) and $k = 3$ (h). Blue and red regions corresponds to the four number of solutions for the IKP for $\det(\mathbf{A}) > 0$ and $\det(\mathbf{A}) < 0$, respectively.	45
II.13 Slice of the joint space for $\rho_1 = 3$	46
II.14 Slice of the joint space for OM^1 for different values of ρ	46
II.15 Cusp points for OM^1 and OM^2	46
II.16 Slice of the joint space for $\rho_1 = 1$, $k = 0.5$ (a), $\rho_1 = 1$, $k = 1$ (b), $\rho_1 = 1$, $k = 1.5$ (c), $\rho_1 = 2$, $k = 0.5$ (d), $\rho_1 = 2$, $k = 1$ (e), $\rho_1 = 2$, $k = 1.5$ (f), $\rho_1 = 3$, $k = 0.5$ (g), $\rho_1 = 3$, $k = 1$ (h) $\rho_1 = 3$, $k = 1.5$ (i), where the DKP admits four, eight, twelve and sixteen real solutions in the blue, red, yellow and pink region, respectively.	47
II.17 Slice of the joint space for OM^2 for different values of ρ	48
III.1 Projection of parallel singularity curve in workspace	53
III.2 Projection of parallel singularity curve in jointspace	54
III.3 Projection of serial singularity curve in jointspace	55
III.4 Projection of serial singularity curve in workspace	55
III.5 Singularity curves and its 2D projection for $q_1 = 0$	58
III.6 Singularity curves and its 2D projection for $q_4 = 0$	58
III.7 Projection of parallel singularities in projection space $[p_x, p_y, p_z]$ for OM^1 . Variation in the parallel singularities surface for different design parameter $k = 0.5$ (a), $k = 1$ (b), $k = 2$ (c) and $k = 3$ (d), top view of the parallel singularities surface, $k = 0.5$ (e), $k = 1$ (f), $k = 2$ (g) and $k = 3$ (h).	59
III.8 Projection of parallel singularities in projection space $[p_x, p_y, p_z]$ for OM^2 . Variation in the parallel singularities surface for different design parameter $k = 0.5$ (a), $k = 1$ (b), $k = 2$ (c) and $k = 3$ (d), top view of the parallel singularities surface, $k = 0.5$ (e), $k = 1$ (f), $k = 2$ (g) and $k = 3$ (h).	60
III.9 Aspects for OM^1 and OM^2	61
III.10 Singularity curves and characteristic surfaces with $z = 3$ for OM^1 (a) OM^2 (b) .	62
III.11 Slice of an aspect for $z = 3$ and OM^1 for $\det(\mathbf{A}) > 0$ and $\det(\mathbf{A}) < 0$	63
III.12 Slice of an aspect for $z = 3$ and OM^2 for $\det(\mathbf{A}) > 0$ and $\det(\mathbf{A}) < 0$	64
III.13 Basic regions of OM^1 for $\det(\mathbf{A}) > 0$	64
III.14 Basic regions of OM^1 for $\det(\mathbf{A}) < 0$	65
III.15 Basic regions of OM^2 for $\det(\mathbf{A}) > 0$	66
III.16 Basic region of OM^2 for $\det(\mathbf{A}) < 0$	67
III.17 Slice of an aspect for $z = 3$ and $\det(\mathbf{A}) > 0$ for OM^1 (a) OM^2	68
III.18 Variation of $\det(\mathbf{A})$ along trajectory for OM^1 and OM^2	68
III.19 Projection in Q of the trajectories with the cusp curves	69
IV.1 Architecture of an Orthoglide including two assembly modes (a) Workspace plot with $\chi(\rho)$ joint constraints (b) and Parallel singularity $\det(\mathbf{A})$ projected in the workspace as $\xi(\mathbf{X})$ (c) and joint space plot with $\chi(\rho)$ joint limits (d) . . .	76
IV.2 A comparison between the parallel singularity surface for an Orthoglide computed with the joint limits $\mu(\mathbf{X})$ (a) without $\mu(\mathbf{X})$ (b)	77
IV.3 Position of Trajectories 1&2 (a) Trajectory 3 (b) in the workspace of an Orthoglide	78

IV.4	A pictorial representation of the mapping of trajectories from workspace to joint space. Eight different pairs of trajectories, $\Upsilon_1(\rho, \mathbf{t})$ & $\Upsilon_2(\rho, \mathbf{t})$ in joint space are the image of corresponding $\phi_1(\mathbf{t})$ & $\phi_2(\mathbf{t})$. These eight different trajectories are associated with the eight working modes of the Orthoglide. Only one trajectory lies inside the joint space boundary due to to the joint constraints.	79
IV.5	Mapping of the Trajectory 3 from workspace to joint space. Eight different possible solutions of $\Upsilon_3(\rho, \mathbf{t})$ are marked in joint space which are the image of $\phi_3(\mathbf{t})$. There is only one feasible trajectory (marked as 1) lies inside the joint space boundary.	80
IV.6	Representation of the trajectories with singularity surface. Trajectory 1 cuts the parallel singularity surface $\xi(\mathbf{X})$ in four points s_1, s_2, s_3 and s_4 in the workspace of the Orthoglide. Also, it can be seen that Trajectory 2 lies inside $\xi(\mathbf{X})$ as $\mu_2(\mathbf{t}) \neq \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi]$	82
IV.7	Representation of the real singular points s_3 and s_4 along with the singularity surface $\xi(\mathbf{X})$, which is associated with one working mode.	83
IV.8	Variation of $\mu_1(\mathbf{t})$ along Trajectory 1. s_1, s_2, s_3 and s_4 represents the four solutions of $\mu_1(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi]$	83
IV.9	Variation of $\det(\mathbf{A})$ along the Trajectory 1. s_3 and s_4 represents the two solutions of $\det(\mathbf{A}) = 0 \quad \forall t \in [-\pi, \pi]$	84
IV.10	Variation of $\mu_2(\mathbf{t})$ along the Trajectory 2. There are no solutions for $\mu_2(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi]$	85
IV.11	Variation of $\mu_3(\mathbf{t})$ along the Trajectory 3. There are no solutions for $\mu_3(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi]$	85
IV.12	The NaVARo robot	86
IV.13	The virtual model of the NaVARo robot	87
IV.14	The NaVARo transmission system	87
IV.15	The graphic user interface of the NaVARo robot	89
IV.16	Joint space of 3RPR parallel mechanism (a) without joint limits (b) with joint limits. The Red and Blue regions represents the two and four real solutions for the DKP, respectively.	90
IV.17	First assembly mode of the NaVARo shown with pink colored mobile platform and the second assembly mode shown with yellow colored mobile platform for $\theta_1 = -0.585, \theta_2 = -2.680$ and $\theta_3 = 1.508$	91
IV.18	Flow chart of the control loop	92
IV.19	Definition of the angles used to solve the inverse kinematics	93
V.1	Semi industrial prototype of the Orthoglide 5-Axis	99
V.2	Simplified architecture of the Orthoglide for the simulation and the analysis. . .	100
V.3	Location of the trajectories $\phi_i(t)$ along with the parallel singularity surfaces $\xi(\mathbf{X})$. CAD algorithm is used to plot these surfaces for a single working mode. .	102
V.4	Cartesian velocities along the trajectories $\phi_i(t)$	103
V.5	Cartesian acceleration along the trajectories $\phi_i(t)$	103
V.6	Joint parameters values along trajectories $\phi_1(t), \phi_2(t), \phi_3(t), \phi_4(t), \phi_5(t)$ Joint positions (a_i) Joint velocities (b_i) Joint accelerations (c_i), where $i = 1, 2, 3, 4, 5$ which represents different trajectories	104
V.7	Joint errors in $\Delta\rho_1$ (a) $\Delta\rho_2$ (b) and $\Delta\rho_3$ (c) along the trajectories $\phi_i(t)$, Experimental data from the Orthoglide 5-axis	107

V.8	Joint error $\Delta\rho_i$ and proposed error model $\Delta\rho_i^m$ value along trajectory $\phi_1(t)$: $[-80, -80, -140]$, $\phi_2(t)$: $[-80, -80, -80]$, $\phi_3(t)$: $[00, 00, 00]$, $\phi_4(t)$: $[80, 80, 140]$, $\phi_5(t)$: $[80, 80, 140]$ $\Delta\rho_1$ (a_i) $\Delta\rho_2$ (b_i) $\Delta\rho_3$ (c_i), where $i = 1, 2, 3, 4, 5$ which represents different trajectories	108
V.9	Cartesian errors in \mathbf{x} (a) \mathbf{y} (b) and \mathbf{z} (c) along the trajectories $\phi_i(t)$, Image of the joint errors in the workspace of the manipulator	110
V.10	Different starting points on the trajectory to analyse the influence on the accuracy of the Orthoglide.	112
V.11	Joint error values, $\Delta\rho_j$ from experimental data for different starting position [$T_i : [0]$ (in blue), $T_i : [\pi/2]$ (in green), $T_i : [\pi]$ (in red), $T_i : [3\pi/2]$ (in black)] of Orthoglide 5-axis along trajectories $\phi_1(t)$, $\phi_2(t)$, $\phi_3(t)$, $\phi_4(t)$, $\phi_5(t)$ for ρ_1 (a) ρ_2 (b) ρ_3 (c), where $i = 1, 2, 3, 4, 5$ which represents different trajectories and $j = 1, 2, 3$	113
V.12	Cartesian velocities, $\dot{\rho}_i$ (a_i) and Cartesian accelerations, $\ddot{\rho}_i$ (b_i) along trajectory $\phi_5(t)$ for different starting positions [$T_5 : [0]$ (in blue), $T_5 : [\pi/2]$ (in green), $T_5 : [\pi]$ (in red), $T_5 : [3\pi/2]$ (in black)].	116
V.13	Cartesian errors in \mathbf{x} (a_1), \mathbf{y} (a_2), \mathbf{z} (a_3) and norm (a_4) along the trajectories $\phi_5(t)$ for different starting positions [$T_1 : [0]$ (in blue), $T_2 : [\pi/2]$ (in green), $T_3 : [\pi]$ (in red), $T_4 : [3\pi/2]$ (in black)], Image of the joints errors in the workspace of the manipulator.	117
V.14	Cartesian errors, norm, along trajectories $\phi_1(t)$ (a_1), $\phi_2(t)$ (a_2), $\phi_3(t)$ (a_3), $\phi_4(t)$ (a_4) and $\phi_5(t)$ (a_5) for different starting positions [$T_1 : [0]$ (in blue), $T_2 : [\pi/2]$ (in green), $T_3 : [\pi]$ (in red), $T_4 : [3\pi/2]$ (in black)], Image of the joints errors in the workspace of the manipulator.	117
VI.1	Parallel singularity surface of 3RPS parallel robot (a) The methodology can be extended to find the optimized path between point p_1 and p_2 (b) which is shown in red color.	123
VI.2	The virtual model of the 3RRR robot	124

List of Tables

I.1	Description of the fields of CreateManipulator function	13
I.2	Description of the fields of CellDecompositionPlus function	15
I.3	Description of the fields of DiscriminantVariety function	27
I.4	Description of the fields of CellDecomposition function	29
II.1	Definition of the Workspace with CAD	40
II.2	Definition of the Jointspace with CAD	40
III.1	Comparison of the different parameters associated with the parallel singularities for the robots	56
III.2	Comparison of the different parameters associated with the serial singularities for the robots	57
III.3	Solutions of the DKP for $\det(\mathbf{A}) > \mathbf{0}$	68
IV.1	Singular postures on Trajectory 1	84
IV.2	The eight actuation modes of the NaVARo	88
V.1	The Minimum, Maximum and mean absolute values of joint errors along the trajectories	106
V.2	Maximum and mean absolute values of Cartesian position errors	109
V.3	The Minimum, Maximum and mean absolute values of joint errors along the trajectories with different starting position	114
B.1	Different Functions for Analysing in SIROPA	139
B.2	Different Functions for Modeling in SIROPA	140
B.3	Different Functions for Plotting in SIROPA	140
B.4	Different Functions for Base Algebra	146

B.5 Different Functions for Extra Algebra	147
B.6 Different Functions for Trigonometric	147

Nomenclature

ASME	American Society of Mechanical Engineers
IFTToMM	International Federation for the Promotion of Mechanism and Machine Science
PM	Parallel Manipulator(s)
PID	Proportional Integral Derivative
CAD	Cylindrical Algebraic Decomposition
3T	Three Translational Motion
2R1T	Two Rotational and one Translational Motion
IKP	Inverse Kinematics problem
DKP	Direct Kinematics problem
det	Determinant
OM	Operation Mode
R	Revolute Joint
P	Prismatic Joint
S	Spherical Joint
WA	Aspect of the Operation mode
S_c	Characteristic surfaces
WAb	Basic Regions
QA	Basic Components
ρ	Actuated Joint Variables
X	Pose Variables
Ψ	System of Equations
Υ	Projections in Joint space
A	Direct kinematics Jacobian matrices
B	Inverse kinematics Jacobian matrices
$\xi(\mathbf{X})$	Projections of the parallel singularities in the Workspace
$\varepsilon(\rho)$	Projections of the parallel singularities in the joint space
$\chi(\rho)$	Joint limits of the actuator
$\mu(\mathbf{X})$	Projection of joint limits of the actuator in workspace

Introduction

Motivation

Robots have been a reality on factory assembly lines for over twenty years. A revolution in the field of robotics transforms the global economy for past few years, cutting the costs of services for the people. The researchers and engineers across the world made remarkable contributions for the efficient use of robots in the manufacturing, health, entertainment and aeronautical sectors. This doctoral thesis focuses on the different aspects which are associated with efficient planning of desired tasks for parallel robots. These different aspects are mainly categorized in four parts, namely: workspace and joint space analysis, uniqueness domains, trajectory planning and accuracy analysis. The workspace and joint space analysis differentiate the regions with different number of inverse kinematic solutions and direct kinematic solutions using a cylindrical algebraic decomposition algorithm, respectively. The influence of design parameters and joint limits on the workspace boundaries for the parallel robots are reported. Gröbner based elimination methods are used to compute the parallel and serial singularities of the manipulator under study. The descriptive analysis of a family of delta like robots is presented by using algebraic tools to induce an estimation about the complexity in representing the singularities in the workspace and the joint space. The generalized notions of aspects and uniqueness domains are defined for the parallel robot with several operation modes. The characteristic surfaces are also computed to define the uniqueness domains in the workspace. An algebraic method is proposed to check the feasibility of any given trajectory in the workspace to address the well known problem which arises when there exists a singular configuration between the two poses of the end-effectors while discretizing the path with a classical approach. A Framework for the control loop of a parallel robot with several actuation modes is presented, which uses only the inverse geometric model. The accuracy analysis focuses on the estimation of errors in the pose of the end effector due to the joint's errors produced by the PID control loop. The proposed error model, which is based on the static and dynamic properties of the Orthoglide, helps in estimating the error in the Cartesian workspace.

The workspace can be defined as the volume of space of the complete set of poses which the end-effector of the manipulator can reach. Many researchers published several works on the problem of computing these complete sets for robot kinematics. Based on the early studies [1, 2], several methods for workspace determination have been proposed, but many of them are applicable only for a particular class of robots. The workspace of parallel robots

mainly depends on the actuated joint variables, the range of motion of the joints and the mechanical interferences between the bodies of mechanism. There are different techniques based on geometric [3, 4], discretization [5, 7, 8], and algebraic methods [9, 10, 11, 12] which can be used to compute the workspace of parallel robot. The main advantage of the geometric approach is that, it establish the nature of the boundary of the workspace [19]. Also it allows to compute the surface and volume of the workspace while being very efficient in terms of storage space, but when the rotational motion is included, it becomes less efficient.

Interval analysis based methods can be used to compute the workspace but the computation time depends on the complexity of the robot and on the requested accuracy [8]. Discretization methods are usually less complicated and can easily take into account all kinematic constraints, but they require more space and computation time for higher resolutions. The majority of numerical methods which is used to determine the workspace of parallel manipulators includes the discretization of the pose parameters for computing workspace boundaries [7]. There are other approaches, which are based on optimization algorithms [20] for fully serial or parallel manipulators, analytic methods for symmetrical spherical mechanisms [21].

Algebraic methods are used in [12, 11, 14, 22, 23, 24] to study planar or spatial parallel robots. Two main steps are necessary to perform the workspace and joint space analysis. First, the discriminant variety is computed to characterize the boundaries of the workspace and joint space as well as the singularities. Second, the Cylindrical Algebraic Decomposition is used to define the connected regions where there exists a constant number of real solutions to the inverse or direct kinematic problem and no parallel or serial singularities [12, 14, 15]. A cylindrical algebraic decomposition based method is illustrated in [11, 16], which is used to model the workspace and joint space for the 3 RPS parallel robot. The workspace analysis for the 3RPS and delta-like family parallel manipulators are presented in a chapter of the thesis. The cylindrical algebraic decomposition method and Gröbner based computations are used to model the workspace in 3T & 2R1T projection spaces, where the orientation of the mobile platform is represented by using quaternions. A certified three dimensional plotting is proposed to study the shape of the workspace for different delta like manipulators. This thesis presents the results which are obtained by applying algebraic methods for the workspace and joint space analysis of a family of delta-like robot including complexity information for representing the singularities in the workspace and the joint space. The CAD algorithm is used to study the workspace and joint space, and a Gröbner based elimination process is used to compute the parallel and serial singularities of the manipulator.

For the design or the trajectory planning, the workspace of the parallel manipulator is divided into singularity-free regions [17]. The singularities divide the workspace into aspects and the characteristic surfaces induce a partition of each aspect into a set of regions (the basic regions) [18]. For the parallel robots with several inverse and direct kinematic solutions, the aspects are defined as the maximal singularity-free sets in the workspace or in the

cross-product of the joint space by the workspace. An assembly mode is associated with a solution for the Direct Kinematic Problem and a working mode for the Inverse Kinematic Problem (IKP). Practically, a change of assembly mode may occur during the execution of a trajectory between two configurations in the workspace which are not necessarily associated with the same input for a given working mode. The uniqueness domains can be defined as the maximal regions of the workspace where all the displacements of the end-effector can be accomplished without changing of assembly mode and working mode [12].

One of the crucial steps in the trajectory planning is to check the singularity-free paths in the workspace for the parallel manipulators. It becomes a necessary protocol to validate the trajectory when the parallel robot is used in practical applications such as precise manufacturing operations. A trajectory verification problem is presented in [25], based on some validity criteria like whether the trajectory lies fully inside the workspace of the robot and is singularity-free. Singularity-free path planning for the Gough-Stewart platform with a given starting pose and a given ending pose has been addressed in [26] using the clustering algorithm is presented in [27]. An exact method and an approximate method are described in [28] to restructure a path close to the singularity locus into a path that avoids it while remaining close to the original path. Due to the geometry of the mechanism, the workspace may not cover fully the space of poses [27], hence it is necessary to analyze the workspace of the manipulator. A procedure to automatically generate the kinematic model of parallel mechanisms which further used for singularity free path planning is reported in [30]. An algorithm for computing singularity-free paths on closed-chain manipulators is presented in [31], also this method attempts to connect the given two non-singular configurations through a path that maintains a minimum clearance with respect to the singularity locus at all points.

The main drawback of numerical or discretization methods is that there might be a singular configuration between two poses of the end-effectors when discretizing the path. This thesis illustrates a technique based on some algebraic methods to check the feasibility of any given trajectory in the workspace : it allows to write the Jacobian of the manipulator as a function of the time and to check whether its determinant vanishes between two poses. Also, when the trajectory meets a singularity, its location can also be computed.

The accuracy to reach the exact pose for the parallel manipulator for a given trajectory depends on the static and dynamic parameters associated with the manipulator. Due to their better dynamic properties, high load-carrying capacity, high accuracy and stiffness, closed loop mechanisms are best suited for the medical robotics, high-precision and machine tool design applications. Number of links and passive joints in the closed loop mechanism reduce the accuracy of the manipulator. There are different factors which affect the accuracy of the manipulator, some of them are geometrical deviations of the machine parts during their assembly, mechanism motions, elastic deformations of the links and joints due to the force and thermal expansion [32, 33]. There are several article exists on the effect of manufacturing tolerances on the accuracy of the parallel manipulators [34]. In [35], a forward and inverse

error bound analysis is presented to find the error bound in the pose of the end effector for a Gough-Stewart platform when the joint error bounds are given and vice versa. The sensitivity analysis for a three degrees-of-freedom translational parallel kinematic machine with orthogonal linear joints is reported in [36], they have used linkage kinematic analysis and differential vector method to study the influence of the length variation on the pose of the end-effector.

One of the highly addressed problem associated with the end-effector pose error is the manipulator stiffness, which defines the positioning error due to the external loading while executing a specific task by the manipulator. A non-linear stiffness model for the manipulators with the passive joints is presented in [37]. Pashkevich et al [38] proposed a novel calibration approach for the Orthoglide based on the observations of the manipulator legs parallelism. A larger number of contributions in the literature is available on the influence of the statics than the influence of dynamics on computing the error in the pose of the parallel manipulator [41, 42]. A methodology is presented in [43] to project the trajectories in the joint space using Gröbner based elimination methods. In the final chapter of this thesis, results associated with accuracy analysis which focuses on the estimation of error in the pose of end effector due to the joint errors produced by the PID control loop are presented. The proposed error model which is based on the dynamic properties (joint velocities and acceleration) of the Orthoglide helps in estimating the error in the Cartesian workspace.

Thesis Goal and Research Problems

This thesis focuses on the different aspects which are associated with the planning of a task for the parallel manipulators. Starting from the workspace modeling, this document presents the singularity analysis, an algebraic method to check the feasibility of any given trajectory and finally the accuracy analysis for the parallel manipulators. The chapters of this thesis are based on the following problems:

Problem 1

Model the workspace and study the influence of design parameters and joint limits on the workspace boundaries for the parallel manipulators. Also the change in the workspace due to the different configurations or arrangements of the actuators of the parallel manipulators.

Problem 2

Computation of the singularities and their projections with inequalities in the Cartesian space and joint space. Defining the aspect and uniqueness domain for manipulators with several working, assembly and operation modes.

Problem 3

Feasibility of any given trajectory using algebraic methods which ensure the singularity-free path unlike other classical numerical techniques.

Problem 4

Influence of static and dynamic parameters in computing the error in the pose of the end effector. Dependency of the different parameters on the accuracy of the parallel manipulators.

Thesis Structure

This doctoral dissertation includes mainly five chapters and the objectives of this doctoral dissertation are stated as follows:

Chapter 1 presents the definitions of the basic terminologies and the mathematical tools, which are used in the chapters of this thesis. The working modes, assembly modes, aspects and uniqueness domains are illustrated using different examples, based on the available articles. Also the aspects for parallel manipulator with single IKS, multiple IKS and operation modes are discussed. The CAD algorithm is used to study the workspace and joint space, and a Gröbner basis elimination process is used to compute the parallel and serial singularities of the manipulator. A brief description of these mathematical tools is also presented in the later sections of this chapter.

Chapter 2 covers the workspace and joint space analysis for the 3RPS and delta-like family parallel manipulators. Also, the influence of the design parameters and joint limits on the workspace boundaries for the parallel manipulators are reported. The cylindrical algebraic decomposition method and Gröbner basis computations are used to model the workspace in 3T & 2R1T projection spaces, where the orientation of the mobile platform is represented using quaternions. A certified three dimensional plotting is proposed to study the shape of the workspace for different delta like manipulators.

Chapter 3 presents the singularity analysis for the previously defined mechanisms. The Gröbner basis elimination method is used to compute the projection of the singularities in the Cartesian space and in the joint space. The descriptive analysis of a family of delta like robots is presented by using algebraic tools to induce an estimation about the complexity in representing the singularities in the workspace and the joint space. The generalized notions of the uniqueness domains is also presented for the parallel robot with several operation modes. The effect of joint limits on these singularity surfaces are also presented in the later sections of this chapter.

Chapter 4 is devoted to the joint space analysis and to an algebraic method to check the feasibility of any given trajectories in the workspace. The solutions of the polynomial equations associated with the trajectories are projected in the joint space using Gröbner basis elimination methods and the remaining equations are expressed in a parametric form where the articular variables are functions of the time \mathbf{t} unlike any numerical or discretization method. These formal computations allow to write the Jacobian of the manipulator as a function of the time and to check if its determinant can vanish between two poses.

Another benefit of this approach is to use a larger workspace with a more complex shape than a cube, a cylinder or a sphere.

Chapter 5 reports the influence of static and dynamic parameters in computing the error in the pose associated with the trajectory planning made and analyzed with the Orthoglide 5-axis. An error model based on the joint parameters (velocity and acceleration) and experimental data coming from the Orthoglide 5-axis is proposed. Newton and Gröbner based elimination methods are used to project the joint error in the workspace to check the accuracy/error in the Cartesian space. For the analysis, five similar trajectories with different locations inside the workspace are defined using fifth order polynomial equation for the trajectory planning. It is shown that the accuracy of the robot depends on the location of the path as well as the starting and the ending posture of the manipulator due to the acceleration parameters.

Main Contributions

1- Workspace and joint space analysis of the 3-RPS parallel robot

The study of workspace, joint space and singularities together assists the engineers and researchers in the efficient task planning and the selection of the particular configuration of the manipulator for a desired task. This work reports the variations in the workspace, singularities and joint space with respect to the design parameter ' \mathbf{k} ' of the 3-RPS parallel manipulator. The cylindrical algebraic decomposition method and Gröbner based computations are used to model the workspace and joint space with the parallel singularities in 3T & 2R1T projection spaces, where the orientation of the mobile platform is represented by using quaternions. An algorithm is presented to separate the singularity surfaces for the positive and negative values of quaternion q_1 & q_4 for the corresponding operation modes. Depending on the design parameter ' \mathbf{k} ', three different configurations of the 3-RPS parallel manipulator are analyzed.

2- The uniqueness domains for parallel robot with several operation modes and assembly modes

The accurate computation of the workspace and joint space for 3-RPS parallel robotic manipulator is a highly addressed research work across the world. Researchers have proposed a variety of methods to calculate these parameters. In the present context a cylindrical algebraic decomposition based method is proposed to model the workspace and joint space. It is a well know feature that this robot admits two operation modes. We are able to find out a connected set in the joint space with a constant number of solutions for the direct kinematic problem and the locus of the cusp points for the both operation mode. The characteristic surfaces are also computed to define the uniqueness domains in the workspace. A simple 3-RPS parallel with similar base and mobile platforms is used to illustrate this method.

3- Non-singular assembly mode changing trajectories in the workspace for the 3-RPS parallel robot

Having non-singular assembly modes changing trajectories for the 3-RPS parallel robot is a well-known feature. The only known solution for defining such trajectory is to encircle a cusp point in the joint space. In this section, the aspects and the characteristic surfaces are computed for each operation mode to define the uniqueness of the domains. Thus, we can easily see in the workspace that at least three assembly modes can be reached for each operation mode. To validate this property, the mathematical analysis of the determinant of the Jacobian is done. The images of these trajectories in the joint space is depicted with the curves associated with the cusp points.

4- An Algebraic Method to Check the Singularity-Free Paths for Parallel Robots

Trajectory planning is a critical step while programming the parallel robots in a robotic cell. The main problem arises when there exists a singular configuration between two poses of the end-effectors while discretizing the path with a classical approach. This work presents an algebraic method to check the feasibility of any given trajectories in the workspace. The solutions of the polynomial equations associated with the trajectories are projected in the joint space using Gröbner based elimination methods and the remaining equations are expressed in a parametric form where the articular variables are functions of the time t unlike any numerical or discretization method. These formal computations allow to write the Jacobian of the manipulator as a function of the time and to check if its determinant can vanish between two poses. Another benefit of this approach is to use a larger workspace with a more complex shape than a cube, a cylinder or a sphere. For the Orthoglide, a three degree of freedom parallel robot, three different trajectories are used to illustrate this method.

5- Workspace and Singularity analysis of a Delta like family robot

Workspace and joint space analysis are essential steps in describing the task and designing the control loop of the robot, respectively. This section presents the descriptive analysis of a family of delta-like parallel robots using algebraic tools to induce an estimation of the complexity in representing the singularities in the workspace and in the joint space. A Gröbner based elimination is used to compute the singularities of the manipulator and a Cylindrical Algebraic Decomposition algorithm is used to study the workspace and the joint space. From these algebraic objects, we propose some certified three dimensional plotting describing the shape of the workspace and of the joint space which will help the engineers or researchers to decide the most suited configurations of the manipulator they should use for a given task. Also, the different parameters associated with the complexity of the serial and parallel singularities are tabulated, which further enhance the selection of the different configurations of the manipulator by comparing the complexity of the singularity equations.

6- Influence of the trajectory planning on the accuracy of the Orthoglide 5-axis

Usually, the accuracy of parallel manipulators depends on the architecture of the robot, the design parameters, the trajectory planning and the location of the path in the workspace. This work reports the influence of static and dynamic parameters in computing the error in the pose associated with the trajectory planning made and analyzed with the Orthoglide 5-axis. An error model is proposed based on the joint parameters (velocity and acceleration) and on experimental data coming from the Orthoglide 5-axis. Newton and Gröbner based elimination methods are used to project the joint errors in the workspace and to check the accuracy/error in the Cartesian space. For the analysis, five similar trajectories with different locations inside the workspace are defined using fifth order polynomial equations for the trajectory planning. It is shown that the accuracy of the robot depends on the location of the path as well as on the starting and the ending posture of the manipulator due to the acceleration parameters.

7- A Framework for the Control Loop of a Parallel Robot with Several Actuation Modes

There have been several research works on reconfigurable parallel manipulators in the last few years. Some robots are reconfigurable in the sense that the position of the anchor points on the moving platform or of the actuated joints can be changed. Some problems may arise when one intends to make a prototype and develop its control algorithms. A reconfigurable planar parallel robot, named NaVARo, is a 3-DOF planar parallel manipulator with eight actuation modes. This part considers a control scheme of NaVARo while taking an advantage of multiple sensors such as motor encoders, additional absolute encoders and magnetic sensors; which are used to determine the current assembly mode of the manipulator. A methodology is presented to determine the home configuration of the NaVARo.

I

Definitions and Mathematical Tools

Chapter 1 presents the definitions of the basic terminologies and the mathematical tools, which are used in the chapters of this thesis. The working modes, assembly modes, aspects and uniqueness domains are illustrated using different examples, based on the available articles. Also the aspects for parallel manipulator with single IKS, multiple IKS and operation mode are discussed. The cylindrical algebraic decomposition method is used to study the workspace and joint space, and a Gröbner based elimination process is used to compute the parallel and serial singularities of the manipulator. A brief description of these mathematical tools is also presented in the later sections of this chapter.

1.1 Parallel Robots

A parallel robot is a mechanical system with a closed-loop kinematic chain mechanism whose end-effector is linked to the base by several independent kinematic chains. Parallel robots can be categorized in two different type as fully parallel and non-fully parallel manipulators based on the relation between the number of chains and degree of freedom of the end-effector.

Definition 1 *A fully parallel manipulator is a mechanism that includes as many elementary kinematic chains as the mobile platform does admit degrees of freedom. Moreover, every elementary kinematic chain possesses only one actuated joint (prismatic, pivot or kneecap). Besides, no segment of an elementary kinematic chain can be linked to more than two bodies [47].*

Further fully parallel manipulators can be categorized in planar robots (three degrees of freedom in the plane), and spatial robots, which do not move just within a plane. A fully parallel planar manipulator has an end-effector with three degrees of freedom, two translations and one rotation. A review on the computation of mobility of mechanisms is presented in [6].

The parallel architecture provides high rigidity and high payload-to-weight ratio, high accuracy, low inertia of moving parts, high agility, and simple solution for the inverse kinematics problem. The fact that the load is shared by several kinematic chains results in high

payload-to-weight ratio and rigidity. The disadvantages of the parallel manipulators are limited work volume, low dexterity, complicated direct kinematics solution, and singularities that occur both inside and on the envelope of the work volume.

1.1.1 Create Manipulator with SIROPA

SIROPA library Provides modeling, analyzing and plotting functions for different manipulators. We will use the function *CreateManipulator()* of SIROPA library in MAPLE software to create the manipulator virtually, so that we can do the analysis of different parameter associated with the manipulator. Listing 1.1 shows the code architecture of the function:

```

1 CreateManipulator := proc (
2   sys[c]      :: list({ algebraic , algebraic=algebraic , algebraic<algebraic })
3   cart[c]     :: list(name) ,
4   arti[c]    :: list(name) ,
5   passive[o] :: list(name) ,
6   geompars[c]:: list(name) ,
7   spec[o]    :: { list , set }(name=algebraic) ,
8   plotrange[o]:: list(name=range) ,
9   points[p]  :: list(name=list(algebraic)) ,
10  loops[p]   :: list(list(name)) ,
11  chains[p]  :: list(list(name)) ,
12  actuators[p]:: list({ list(name) , name}) ,
13  model[o]   :: string                := "No name" ,
14  precision[o]:: integer              := 4 ,
15  {
16  noradical  :: truefalse             := false
17  }
18 )

```

Listing I.1 – Architecture of Create Manipulator

In the Listing 1.1, there are some compulsory input for the function which are marked as [c], and rest which are marked as [o], are optional to create the manipulator virtually for further analysis. *Points, loops, chains and actuators*, these are the input parameters to create the plot of the manipulator. The pose variables are the essential input parameters to define the mechanism. The input parameter *sys* is the set of constraint equations associated with the motion of manipulator. These constraint equations can be in the form of Euler angle representation or quaternions [11].

This function returns a data structure of type Manipulator containing the fields, which is given in Table I.1. Further, the values of these fields can be retrieved or changed according to the analysis to be performed. Brief description of these fields is presented in the Table I.1.

1.1.2 Workspace and Jointspace

Workspace in general can be defined as the volume of space which the end-effector of the manipulator can reach. The workspace of parallel robots mainly depends upon the actuated joint variables, the range of motion of the joints and the mechanical interferences between the bodies of mechanism. Below is the different types of workspace based on the constraint impose on the parameters [72, 66, 69]:

Equations	a list of polynomials [p1, ..., pk]: the modelling equations
Constraints	a list of strict inequalities: the constraint inequalities
PoseVariables	a list of names: the variables defining the pose
ArticularVariables	a list of names: the control parameters
PassiveVariables	a list of names: the remaining variables
GeometricParameters	a list of names: the geometric parameters
GenericEquations	a list of polynomials: the modelling equations with symbolic geometric parameters
GenericConstraints	a list of strict inequalities: the constraint inequalities appearing in sys
Precision	an integer: the number of correct digits
PoseValues	the pose values substituted in the GenericEquations to get the Equations
ArticularValues	the articular values substituted in the GenericEquations to get the Equations
PassiveValues	the passive values substituted in the GenericEquations to get the Equations
GeometricValues	the geometric values substituted in the GenericEquations to get the Equations
DefaultPlotRanges	ranges used by default for plotting if provided
Points	the points coordinate of the robot
Loops	the frame loops of the robot
Chains	the frame chain of the robot
Actuators	the actuators of the robot
Model	a string: the name of the modelling

Table I.1 – Description of the fields of CreateManipulator function

- **Constant orientation workspace** or **Translation workspace**: all possible locations of the end-effector of the robot that can be reached with a given orientation.
- **Orientation workspace**: all the possible orientations that can be reached while end-effector is in a fixed location.
- **Maximal workspace** or **Reachable workspace**: all the locations of end-effector that may be reached with at least one orientation of the platform.
- **Inclusive orientation workspace**: all the locations of end-effector that may be reached with at least one orientation among a set defined by ranges on the orientation angles. The maximal workspace is a particular case of inclusive orientation workspace for which the ranges for the orientation angles are $[0, 2\pi]$.
- **Total orientation workspace**: all the locations of end-effector that may be reached with all the orientations among a set defined by ranges on the orientation angles.
- **Dextrous workspace**: all the locations of end-effector for which all orientations are possible. The dextrous workspace is a particular case of total orientation workspace, the ranges for the rotation angles being $[0, 2\pi]$.
- **Complementary workspace**: it is defined by the variables of workspace as well as jointspace, and helps in defining the trajectory.

There are different techniques based on geometric, discretization, numerical and algebraic methods which are used to calculate the workspace of parallel robot. The main advantage of the geometric approach is that, it establish the nature of the boundary of the workspace [19]. Also it allows the computation of the surface and volume of the workspace while being very efficient in terms of storage space, but if the rotational motion is included, it becomes more complex. The interval analysis based method can be used to compute the workspace but the computation time depends on the complexity of the robot and the accuracy requested. The ALIAS library is a good implementation for the parallel robots [8].

```

1 CellDecompositionPlus := proc (
2 equ          :: list(algebraic),
3 ineq         :: list(algebraic),
4 vars        :: list(name),
5 pars        :: list(name) := [
6             op(indets([equ, ineq], name)
7             minus {op(vars)}) ],
8 {
9 nofactor     :: truefalse := false,
10 gbfactor    :: truefalse := false,
11 noalrootstest :: truefalse := false
12 }
13 )

```

Listing I.2 – Architecture of CellDecompositionPlus

equ	a list of polynomials and trigonometric expressions: the equations.
ineq	a list of polynomials and trigonometric expressions: the inequalities where each & expression \mathbf{p} stands for $\mathbf{p} > \mathbf{0}$.
vars	a list of names: the variables of the system
pars[o]	a list of names: the parameters of the system; default value: the remaining variables of equ and ineq

Table I.2 – Description of the fields of CellDecompositionPlus function

Discretization methods are usually less complex and take into account all kinematic constraints, but require more space and computation time for higher resolutions. The majority of numerical methods which is used to determine the workspace of parallel manipulators includes the discretization of the pose parameters for the determination of workspace boundaries[7].

The workspace and joint space analysis differentiate the regions with different number of inverse kinematic solutions and direct kinematic solutions using a cylindrical algebraic decomposition algorithm, respectively. The CellDecompositionPlus function, shown in Listing 1.2, decomposes the parameter space of a parametric polynomial system into cells in which the original system has a constant number of solutions. This function returns a data structure the same as the one returned by the maple function RootFinding[Parametric][CellDecomposition][52, 56]. The main difference is that it handles trigonometric expressions. The function returns a data structure that can be used for plotting the regions of the parameter space for which the system has a given number of solutions and extracting sample points in the parameter space for which the system has a given number of solutions. The type of the output is a solution record with the following fields: Equations, Inequalities, Filter, Variables, Parameters, DiscriminantVariety, ProjectionPolynomials, and SamplePoints[52, 56].

The input system must satisfy the following properties:

- The number of equations is equal to or greater than the number of indeterminates.
- At most finitely many complex solutions exist for almost all complex parameter values (i.e, the system is generically zero-dimensional).
- For almost all complex parameter values, there are no solutions of multiplicity greater than one (i.e, the system is generically radical); in particular, the input equations are square-free

I.2 Kinematics : IKP and DKP

Kinematics is the science of motion that treats motion without regard to the forces which cause it. Kinematics involves the study of position, velocity, acceleration, and all higher order derivatives of the position variables (with respect to time or any other variables). Hence, the study of the kinematics of manipulators refers to all the geometrical and time-based

properties of the motion. The two very basic problem in the study of mechanical manipulation is direct kinematic problem (DKP) and inverse kinematic problem (IKP). Specifically, given a set of joint angles or parameters, the DKP is to compute the position and orientation of the end-effector relative to the base, also the problem can be posed as: To change the representation of manipulator position from a joint space description into a Cartesian space description. IKP can be posed as: given the position and orientation of the end-effector of the manipulator, calculate all possible sets of joint angles or parameters that could be used to attain this given position and orientation, also the problem of mapping of locations in three-dimensional Cartesian space to locations in the robot's internal joint space can be termed as IKP.

For a manipulator, the relation permitting the connection of input values (\mathbf{q}) with output values (\mathbf{X}) is the following:

$$F(\mathbf{q}, \mathbf{X}) = 0 \quad (\text{I.1})$$

This definition can be applied to serial or parallel manipulators. Differentiating equation I.1 with respect to time leads to the velocity model:

$$\mathbf{A} \dot{\mathbf{X}} + \mathbf{B} \dot{\mathbf{q}} = 0 \quad (\text{I.2})$$

Where,

$$A = \frac{\partial F}{\partial X} \quad \text{and} \quad B = \frac{\partial F}{\partial q}$$

Moreover, $\dot{\mathbf{X}} = [\omega \quad \dot{c}]^T$, where ω and \dot{c} represents the rotational and translational velocities, respectively. \mathbf{A} and \mathbf{B} are respectively the direct-kinematics and the inverse-kinematics matrices of the manipulator. These matrices are useful for the determination of the singular configurations [57].

I.3 Singularities

A singularity is in general a point at which a given mathematical object is not defined, or a point of an exceptional set where it fails to be well-behaved in some particular way. Singularities of a robotic manipulator are important feature that essentially influence its capabilities. Mathematically, a singular configuration may be defined as rank deficiency of the Jacobian describing the differential mapping from the jointspace to the workspace and vice versa. Singular configurations for parallel manipulators are particular poses of the end-effector, for which it loses their inherent infinite rigidity, and in which the end-effector will have uncontrollable degrees of freedom. From equation I.2, \mathbf{A} and \mathbf{B} are the direct kinematics and the inverse kinematics matrices of the manipulator respectively. A singularity occurs whenever $\det(\mathbf{A})$ or $\det(\mathbf{B})$ (or both) vanishes [57, 66]. Depending on these three conditions, singularity can be differentiated into following three types:

- $\det(\mathbf{B}) = 0$: Type1 Singularity or Serial Singularities
- $\det(\mathbf{A}) = 0$: Type2 Singularity or Parallel Singularities

- $\det(\mathbf{A}) = 0$ and $\det(\mathbf{B}) = 0$: Type3 Singularity
- Constraint Singularities

I.3.1 Parallel Singularities

Parallel singularities occur when the determinant of the direct kinematics matrix \mathbf{A} vanishes. The corresponding singular configurations are located inside the workspace. They are particularly undesirable because the manipulator can not resist any force and control is lost. There are mainly two issues associated with parallel singularity, control issues where the pose of the end-effector is no longer controllable and safety issues, where the elements of the robot may be submitted to very large forces, causing a damage to the robot. Using the function *ParallelSingularities()* of SIROPA library in MAPLE software, parallel singularity analysis of different robot can be done efficiently.

```
1 ParalleSingularities := proc ( robot      :: Manipulator )
2 robot                a data structure returned by a function mechanisms.
```

Listing I.3 – Parallel Singularities

Figure I.1 shows the projections of singularity curve in workspace and jointspace. Following are the MAPLE code for computing the parallel singularity curve and its projection in workspace and jointspace:

```
1 s2      := ParalleSingularities ( robot      :: Manipulator )
2 s2_cart := Projection          ( s2, robot:-PoseVariables )
3 s2_art  := Projection          ( s2, robot:-ArticularVariables )
```

Listing I.4 – Projection of parallel singularities in workspace and jointspace

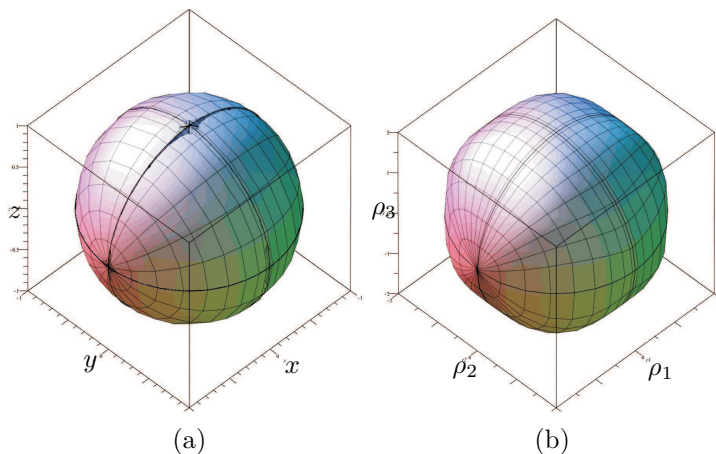


Figure I.1 – Projection of parallel singularity curves of Orthoglide in Workspace (a) and Jointspace (b)

In the Listing 1.4, $\mathbf{s2}$ is the equation associated with the parallel singularity curve, whereas, $\mathbf{s2_cart}$ and $\mathbf{s2_art}$ are the variables associated with the projection of $\mathbf{s2}$ in workspace and jointspace respectively.

1.3.2 Serial Singularities

Serial singularities occur when the determinant of the inverse kinematics matrix \mathbf{B} vanishes. When the manipulator is in such a singularity, there is a direction along which no Cartesian velocity can be produced. serial singularities occur on the boundary of the workspace of a parallel manipulator. Due to serial singularities, the actuated joints velocities reach extreme values. Serial singularity analysis of different robot can be done using the function *SerialSingularities()* in SIROPA library.

```
1 SerialSingularities := proc ( robot      :: Manipulator )
2 robot              a data structure returned by a function mechanisms .
```

Listing I.5 – Serial Singularities

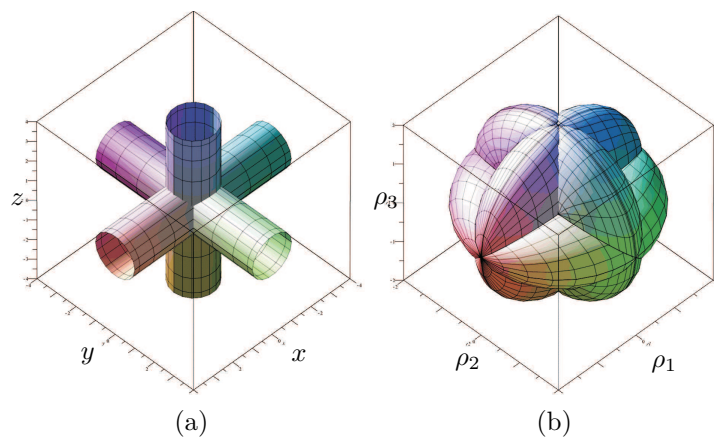


Figure I.2 – Projection of serial singularity curves of Orthoglide in Workspace (a) and Jointspace (b)

Figure I.2 shows the projections of singularity curve in workspace and jointspace. Following are the MAPLE code for computing the parallel singularity curve and its projection in workspace and jointspace:

```
1 s1      := SerialSingularities ( robot      :: Manipulator )
2 s1_cart := Projection        ( s1 , robot:-PoseVariables )
3 s1_art  := Projection        ( s1 , robot:-ArticularVariables )
```

Listing I.6 – Projection of serial singularities in workspace and jointspace

In the Listing 1.6, $s1$ is the equation associated with the serial singularity curve, whereas, $s1_cart$ and $s1_art$ are the variables associated with the projection of $s1$ in workspace and jointspace respectively.

1.4 Assembly modes and Working modes

The algebraic elimination approach for solving the direct kinematics gives several solutions, which refers to the several poses of the end-effector for given values of the joint coordinates. It is therefore possible to assemble the manipulator in different ways, and these different

configurations are known as assembly modes of the manipulator [47]. Similarly, the multiple inverse kinematic solutions induce multiple postures for each leg of the manipulator.

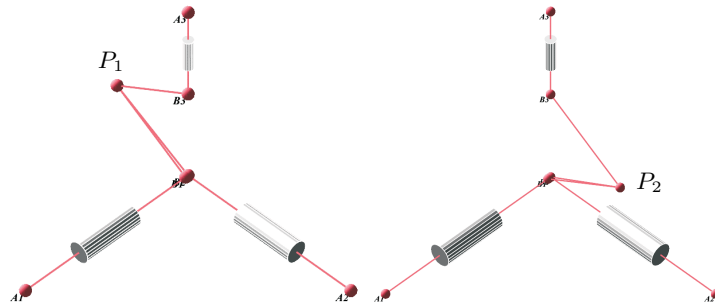


Figure I.3 – Two assembly modes of Orthoglide

Figure I.3 shows the existence of two assembly mode, where p_1 and p_2 are two solutions for DKP at particular joint value.

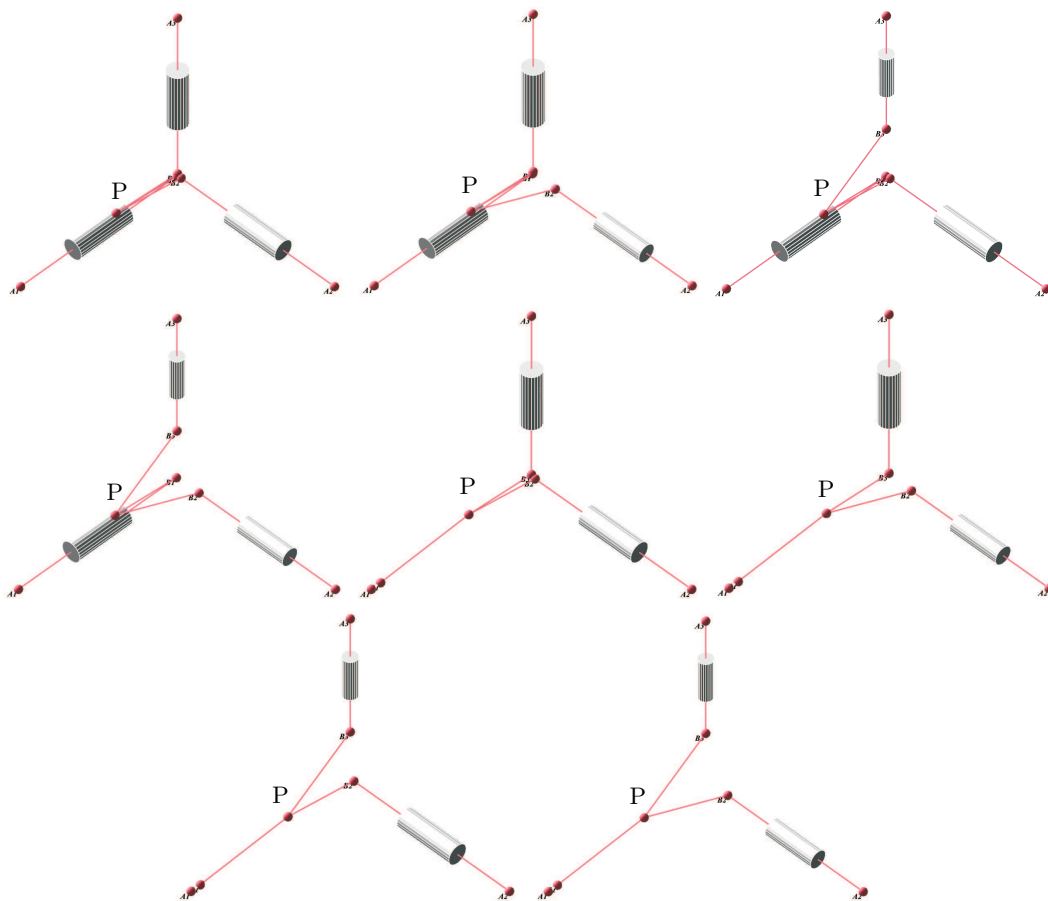


Figure I.4 – Eight posture of orthoglide

Figure I.4 shows the existence of eight different postures for a point p , which is the eight different solutions (joint values) for IKP. A posture changing trajectory is equivalent

to a trajectory between two inverse kinematic solutions and an assembling mode changing trajectory is equivalent to a trajectory between two direct kinematic solutions.

The notion of working modes was introduced in [47] for parallel manipulators with several solutions to the inverse kinematic problem and whose matrix \mathbf{B} is diagonal. A working mode, denoted by Mf_i is the set of robot configurations for which the sign of \mathbf{B}_{jj} does not change and does not vanish. A robot configuration is represented by the vector (\mathbf{X}, \mathbf{q}) [12].

$$Mf_i = \left\{ \begin{array}{l} (\mathbf{X}, \mathbf{q}) \in W \times Q \text{ such that } \text{sign}(B_{jj}) = \text{constant}, \\ \text{for } j = 1, \dots, n \text{ and } \det \mathbf{B} \neq 0 \end{array} \right\} \quad (\text{I.3})$$

Therefore, the set of working modes is obtained using all combinations of sign of each term \mathbf{B}_{jj} . Changing working mode is equivalent to changing the posture of one or several legs. A working mode is defined in $W \times Q$ because the terms of \mathbf{B}_{jj} depend on both \mathbf{X} and \mathbf{q} .

For a working mode Mf_i , we have only one inverse kinematic solution. So, we can define an application that maps \mathbf{X} onto \mathbf{q} :

$$g_i(X) = q \quad (\text{I.4})$$

Then the images in W of a posture \mathbf{q} in Q is denoted by:

$$g_i^{-1}(q) = \left\{ X \mid (X, q) \in Mf_i \right\} \quad (\text{I.5})$$

1.5 Operation Modes

Notion of operation mode (OM) was introduced in [58] to explain the behavior of the DYMO robot. An operation mode is associated with a specific type of motion. For the DYMO robot, there are five operation modes: translational, rotational, planar (two types) and mixed motions. In the workspace W , for each operation mode, W_{OM^j} is defined such that

- $W_{OM^j} \subset W$
- $\forall X \in W_{OM^j}$, OM is constant

For a parallel robot with several operating modes, the pose can be defined by fixing the same number of parameters as the degree of freedom of the mobile platform. Given an operation mode OM^j , if we have a single inverse kinematic solution, we can define an application that maps \mathbf{X} onto \mathbf{q} :

$$g_j(\mathbf{X}) = \mathbf{q} \quad (\text{I.6})$$

Then, the images in W_{OM^j} of a posture \mathbf{q} in the joint space Q are defined by:

$$g_j^{-1}(\mathbf{q}) = (\mathbf{X} \mid (\mathbf{X}, \mathbf{q}) \in OM^j) \quad (\text{I.7})$$

where g_j^{-1} stands to be the direct kinematic problem for the operation mode j .

I.6 Aspects and Uniqueness Domains

The notion of aspect was introduced by [59] to cope with the existence of multiple inverse kinematic solutions in serial manipulators. The notion of aspect was defined for parallel manipulators with single inverse kinematic solution [60] to cope with the existence of multiple direct kinematic solutions. Moreover, the notion of aspect was redefined in [47] for fully parallel manipulators with multiple inverse and direct kinematic solutions.

I.6.1 Aspect for parallel manipulators with single IKS

The aspects WA_i are defined as the maximal sets such that:

- $WA_i \subset W$
- WA_i is connected
- $\forall X \in WA_i, \det(\mathbf{A}) \neq 0$

I.6.2 Aspect for parallel manipulators with multiple IKS

The generalized aspects A_{ij} are defined as the maximal sets in $W.Q$ so that

- $A_{ij} \subset W.Q$
- A_{ij} is connected
- $A_{ij} = (X, q) \in Mf_i \setminus \det(\mathbf{A}) \neq 0$

The projection of the generalized aspects in the workspace yields the parallel aspects WA_{ij} so that

- $WA_{ij} \subset W$
- WA_{ij} is connected

The parallel aspects are the maximal singularity-free domains in the workspace for one given working mode. The projection of the generalized aspects in the jointspace yields the serial aspects QA_{ij} so that

- $QA_{ij} \subset Q$
- QA_{ij} is connected

The serial aspects are the maximal singularity-free domains in the jointspace for one given working mode.

I.6.3 Characteristic surfaces

The characteristic surfaces were introduced in [84] to define the uniqueness domains for serial cuspidal robots. This definition was extended to parallel robots with one inverse kinematic solution in [17] and to parallel robots with several inverse kinematic solutions in [77, 12].

Let WA_{ij} be an aspect in the workspace, W . The characteristic surfaces, $S_c(WA_{ij})$, are defined as the preimage in WA_{ij} of the boundary ∂WA_{ij} that delimits WA_{ij} .

$$S_c(WA_{ij}) = g_i^{-1}(g_i(\partial WA_{ij})) \cap WA_{ij} \quad (\text{I.8})$$

where:

- g_i is defined in Equation I.4
- g_i^{-1} is a notation defined in equation I.5. Let $\mathbf{C} \subset Q$:

$$g_i^{-1}(\mathbf{C}) = \{ \mathbf{X} \in W \setminus g_i(\mathbf{X}) \in \mathbf{C} \} \quad (\text{I.9})$$

The characteristic surfaces coincide with the pseudo-singularities when the robot admits only two aspects in workspace for each working mode [12] and is defined by:

$$S_c(WA_{ij}) = g_i^{-1}(g_i(\partial WA_{ij})) \quad (\text{I.10})$$

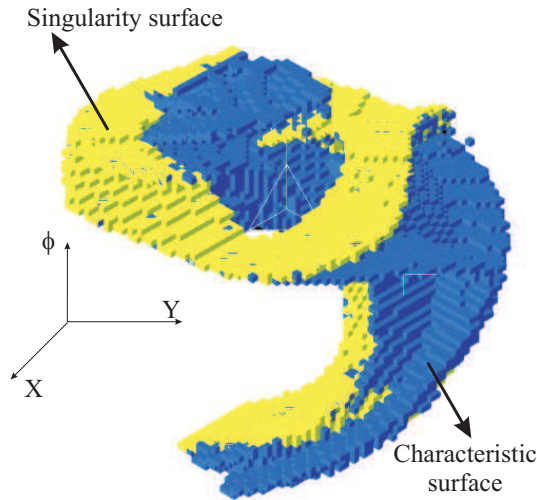


Figure I.5 – Octree model of the singularities and of the characteristic surfaces S_{c1}

Figure I.5 can be taken as the example of characteristic surfaces [60], which depicts singularity surface S along with the characteristic surface S_{c1} of 3-RPR planar parallel manipulator.

1.6.4 Basic components and Basic regions

Let WA_{ij} be an aspect in the workspace, W . The basic regions of WA_{ijk} , denoted WAb_{ijk} , $k \in K$, are defined as the connected components of the set $WA_{ij} \setminus S_c(WA_{ij})$.

The basic regions induce a partition on WA_{ij} :

$$WA_{ij} = (\cup_{k \in K} WAb_{ijk}) \cup S_c(WA_{ij}) \quad (\text{I.11})$$

Figure I.6 can be taken as the example of basic regions [60], which depicts 28 basic regions for 3-RPR planar parallel manipulator. Also in Figure I.7, octree model of the basic components is shown. The coincident components yield domains with 2, 4 or 6 solutions for the direct kinematic problem : the upper one domains contain two coincident basic components, the middle five domains are composed of four coincident domains, and the last two domains contain six coincident basic components. Let $QAb_{ijk} = g(WAb_{ijk})$, QAb_{ijk} is

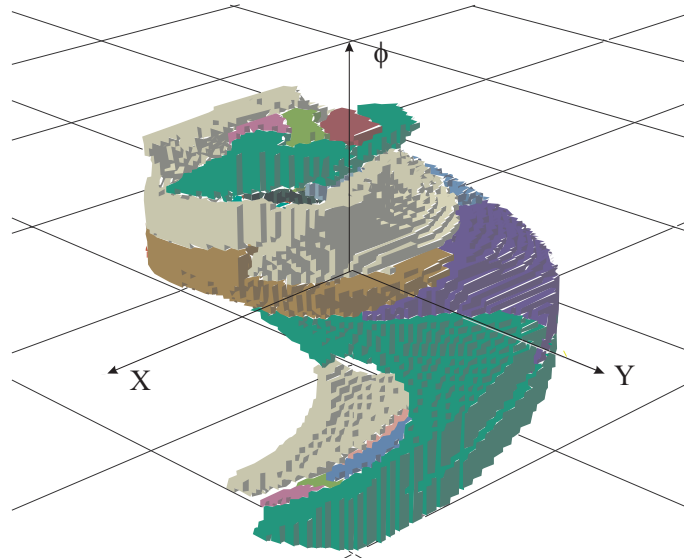


Figure I.6 – Octree model of the basic regions

a domain in the reachable joint space Q called basic components. Let WA_{ij} an aspect in workspace and QA_{ij} its image under g . The following relation holds:

$$QA_{ij} = (\cup_{k \in K} QAb_{ijk}) \cup G(S_c(WA_{ij})) \tag{I.12}$$

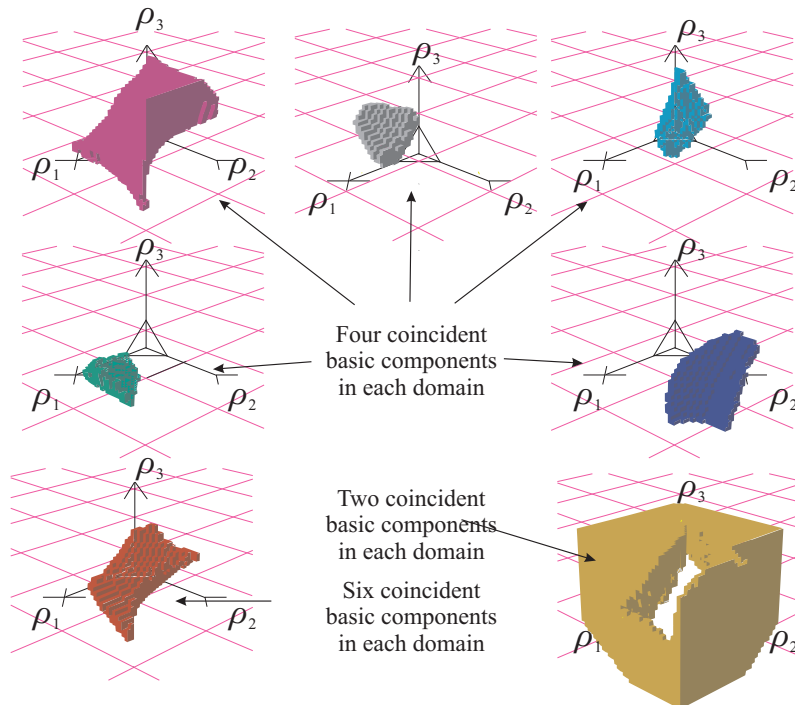


Figure I.7 – Octree model of the basic components

1.6.5 Uniqueness domains

The uniqueness domains Wu_{il} are the union of two sets:

- The set of adjacent basic regions $(\cup_{k \in K} WAb_{ijk})$ of the same aspect WA_{ij} whose respective preimages are disjoint basic components, and
- the set of the characteristic surfaces $S_c(WAb_{ijk})$ $k \in K'$ which separate these basic components:

$$Wu_{il} = (\cup_{k \in K} WAb_{ijk}) \cup S_c(WAb_{ijk}) \quad (I.13)$$

with $K' \subset K$ such that

$$\forall k_1, k_2 \in JK', g(WAb_{ij_1}) \cap g(WAb_{ij_2}) = \emptyset$$

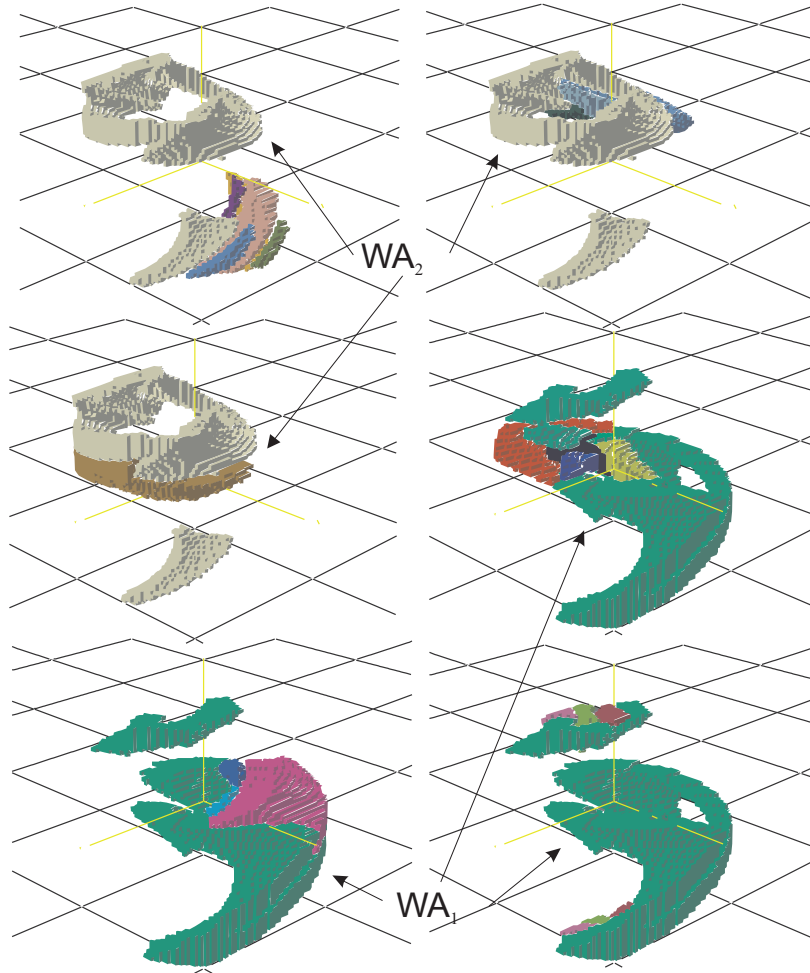


Figure I.8 – Octree model of the six uniqueness domains

Figure I.8 can be taken as the example of uniqueness domain [60], Six uniqueness domains has been found for the 3RPR planar parallel manipulator as shown in Figure I.8.

I.7 Non-Singular assembly mode changing trajectories

Non singular assembly mode change refers that parallel manipulator can pass from one solution of the direct kinematics into another without crossing a singularity. It was first reported in [61]. Earlier, it was shown that to move from one assembly-mode to another, a 3-RPR planar parallel manipulator should cross a singularity [62]. But later it was reported that using numerical experiments, non singular change of assembly mode is possible [61]. The non-singular change of assembly mode in parallel manipulators is often associated with the presence of cusps and the non-singular change of assembly mode is realized by turning around a cusp point, or a cuspidal edge of the singularity surface (see for instance [63, 64, 14, 65]). It has also been reported that non-singular change of assembly modes can be realized by following an 'alpha curve' (i.e. a fold curve intersecting itself transversally) [67, 68], and that the presence of cusps is not necessary for the existence of non singular assembly mode changes [71].

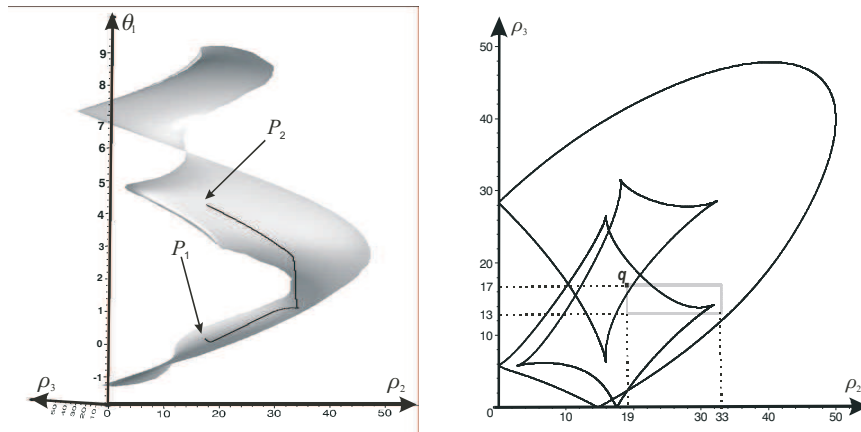


Figure I.9 – Non-singular assembly-mode changing trajectory connecting the two assembly-modes P_1 and P_2

In [18], author investigated the question of whether a change of assembly-mode must occur or not when moving between two prescribed poses in the workspace. They defined the uniqueness domains in the workspace as the maximal regions associated with a unique assembly-mode and proposed a calculation scheme for 3-RPR planar parallel manipulators using octrees. They showed that up to three uniqueness domains exist in each singularity-free region. When the starting and goal poses are in the same singularity-free region but in two distinct uniqueness domains, a non-singular change of assembly-mode is necessary.

Figure I.9 shows a non-singular assembly-mode changing trajectory connecting the two assembly-modes P_1 and P_3 for 3-RPR parallel manipulators and is taken from [70]. To connect the assembly-modes P_1 and P_3 , the trajectory in the joint space encircles two cusp points as shown in Figure I.9. Also in [70], it is verified that any path from P_1 and P_2 must encircle these two cusp points. In [71], it has been shown that a parallel manipulator can execute non-singular changes of assembly modes while its joint space is free of cusp

points and cuspidal edges. The analysis is done on two degrees of freedom manipulator and is derived from a 3-RPR manipulator.

1.8 Standard Bases (Gröebner Bases)

The method of Gröebner bases provides a uniform approach to solving a wide range of problems expressed in terms of sets of multivariate polynomials. The Gröebner basis gives us a method for writing a system of algebraic equations $f(x_1, \dots, x_n) = 0$ in the unknowns x_1, \dots, x_n with finitely many solutions into a system that has the same roots and which is in triangular form $g_n(x_n) = 0, g_{n-1}(x_{n-1}, x_n) = 0, \dots, g_1(x_1, \dots, x_n) = 0$, called a Gröebner basis. There are few drawbacks of Gröebner basis e.g. the calculation time of the Gröebner basis is mainly dependent upon the number of equations and their degree and also its calculation with real numbers is numerically unstable [72].

Groebner basis theory can be used to compute the projections π_Q and π_W . Let P be a set of polynomials in the variables $X = (x_1, \dots, x_n)$ and $q = (q_1, \dots, q_n)$. Moreover, let V be the set of common roots of the polynomial in P , let W be the projection of V on the workspace and Q the projection on the joint space. It might not be possible to represent W (resp. Q) by polynomial equations. Let \bar{W} (resp. \bar{Q}) be the smallest set defined by polynomial equations that contain W (resp. Q) [12]. A Gröebner basis P is a polynomial system equivalent to P , satisfying some additional specific properties. The Groebner basis of a system depends on the chosen ordering on the monomials.

For the projection π_Q , when we choose an ordering eliminating \mathbf{q} , the Gröebner basis of P contains exactly the polynomials defining \bar{W} .

For the projection π_W , when we choose an ordering eliminating \mathbf{X} , the Gröebner basis of P contains exactly the polynomials defining \bar{Q} .

1.9 Discriminant Variety and Cylindrical Algebraic Decomposition

The notion of discriminant variety is a generalization of the discriminant of a univariate polynomial, describing all the critical points of the system, including singularities, solutions of multiplicity greater than one, and solutions at infinity. It is a subset of the parameter space of lower dimension [52, 53].

A discriminant variety has the following property: It divides the parameter space into open, full-dimensional cells such that the number of solutions of the system `sys` is constant for parameter values chosen from the same open cell. The `DiscriminantVariety(sys, vars, pars)` calling sequence computes a discriminant variety of the system `sys` of equations and inequalities with respect to the indeterminates `vars` and the parameters `pars`.

```

1 DiscriminantVariety (sys, vars, pars)
2 DiscriminantVariety (eqs, ineqs, vars, pars)

```

Listing I.7 – Architecture of Discriminant Variety

sys	list of equations and strict inequalities between polynomials with rational coefficients
vars	list of names; the indeterminates
pars	(optional) list of names; the parameters
eqs	list of polynomials f with rational coefficients, representing equations of the form $f = 0$
ineqs	list of polynomials g with rational coefficients, representing constraint inequalities of the form $0 < g$

Table I.3 – Description of the fields of DiscriminantVariety function

The DiscriminantVariety(eqs, ineqs, vars, pars) calling sequence computes a discriminant variety of the system

$$[f = 0, 0 < g]_{f \in eqs, g \in ineqs} \quad (\text{I.14})$$

of equations and inequalities with respect to the indeterminates vars and the parameters pars.

The input system must satisfy the following properties:

- There are at least as many equations as indeterminates.
- At least one and at most finitely many complex solutions exist for almost all complex parameter values (that is, the system is generically solvable and generically zero-dimensional).
- For almost all complex parameter values, there are no solutions of multiplicity greater than one (that is, the system is generically radical). In particular, the input equations are square-free.

An error occurs if one of these conditions is violated.

- The result is returned as a list of lists of polynomials in pars such that the discriminant variety is the union of the set of solutions of the polynomials in each inner list.
- If pars is not specified, it defaults to all the names in sys that are not indeterminates.
- This command will attempt to find a minimal discriminant variety, but it may return a proper superset in the case that it does not succeed.
- The discriminant variety is computed using Gröebner Basis techniques.

Example

```

1 with (RootFinding [ Parametric ])
2 DiscriminantVariety [ a*x^2 = 1, b*z+y = 0, c*z+y = 0, 0 < c ], [ x, y, z ]
3
4 Output          [[ a ], [ c ], [ b-c ]]

```

Listing I.8 – Example of Discriminant Variety

The discriminant variety in this first example is $(a, b, c) : a = 0$ or $b = c$ or $c = 0$. The case $a = 0$ gives a solution of the first equation at infinity. In the case $b = c$, the second

and third equations coincide and therefore the system becomes underdetermined and has infinitely many solutions. Finally, the case $c = 0$ corresponds to a boundary case for the inequality $0 < c$.

A cylindrical algebraic decomposition of the n -dimensional real space is a partition of the whole space into connected semi-algebraic subsets such that the cells in the partition are cylindrically arranged, that is, the projection of any two cells onto any lower dimensional real space is either equal or disjoint. This decomposition is called F -invariant if for any given cell, the sign of each polynomial in F does not change over the cell. `CylindricalAlgebraicDecompose(F, R)` returns an F -invariant cylindrical algebraic decomposition of the n -dimensional real space, where n is the number of variables in R . This assumes that R has characteristic zero and no parameters, such that the base field of R is the field of rational numbers [54].

The output of `CylindricalAlgebraicDecompose(F, R)` has several possible formats controlled by the options `output=piecewise`, `output=tree`, `output=list`, `output=cadcell` and `output=rootof`. In all formats, each cell provides at least two pieces of information, the index of the cell and a sample point of the cell. In the `output=cadcell` and `output=rootof` formats, a defining semi-algebraic system (called a Tarski Formula) is also provided. Due to the cylindricity property, cells can be organized in a hierarchical manner. This is the purpose of `output=piecewise` and `output=tree`, whereas the other three formats are flat representations. Due to the potentially large number of cells, the `output=cadcell` format only shows the name `cadcell` for each cell in the decomposition. However, `cadcell` is a type and an object of that type can be passed to `Display`. It can also be passed to `SamplePoints` in order to access the sample point of the cell. The `output=rootof` format is meant to be compatible with the output format of the `solve` command.

The `CellDecomposition` command decomposes the parameter space of a parametric polynomial system into cells in which the original system has a constant number of solutions [55].

```

1 CellDecomposition (sys, vars, pars, options)
2 CellDecomposition (eqs, posineqs, vars, pars, options)
3 CellDecomposition (eqs, posineqs, nzineqs, vars, pars, options)

```

Listing I.9 – Architecture of Cell Decomposition

The command returns a data structure that can be used for (for example):

- Plotting the regions of the parameter space for which the system has a given number of solutions.
- Extracting sample points in the parameter space for which the system has a given number of solutions.
- Extracting boxes in the parameter space in which the system has a given number of solutions.

The record returned captures information about the solutions of the system depending on the parameter values, including:

- A discriminant variety

sys	list of equations and strict inequalities between polynomials with rational coefficients
vars	list of names; the indeterminates
pars	(optional) list of names; the parameters
eqs	list of polynomials f with rational coefficients, representing equations of the form $f = 0$
posineqs	list of polynomials g with rational coefficients, representing constraint inequalities of the form $0 < g$
nzineqs	list of polynomials g with rational coefficients, representing constraint inequations of the form $g \neq 0$
options	sequence of optional equations of the form keyword=value, where keyword is either output or method

Table I.4 – Description of the fields of CellDecomposition function

- For each full-dimensional open cell, a sample point strictly in the interior of the cell; if possible, the coordinates of the sample point are chosen to be integers

The input system must satisfy the following properties:

- The number of equations is equal to or greater than the number of indeterminates
- At most finitely many complex solutions exist for almost all complex parameter values (that is, the system is generically zero-dimensional)
- For almost all complex parameter values, there are no solutions of multiplicity greater than one (that is, the system is generically radical); in particular, the input equations are square-free

I.10 Conclusions

This Chapter is dedicated to the definitions of the basic terminologies and the mathematical tools, which are used in the chapters of this thesis. The working modes, assembly modes, aspects and uniqueness domains are explained using different examples, based on the available articles. Also the aspect for parallel manipulator with single IKS, multiple IKS and operation mode are discussed. The cylindrical algebraic decomposition method is used to study the workspace and joint space, and a Gröbner based elimination process is used to compute the parallel and serial singularities of the manipulator. A brief description of these mathematical tools is also presented.

II

Workspace and Joint Space Analysis

The accurate calculation of the workspace and joint space for the parallel manipulators is a highly addressed research work across the world. One of the crucial steps in the trajectory planning is to check the singularity-free paths in the workspace, which requires the prior knowledge of the workspace for the parallel manipulators. This chapter covers the workspace and the jointspace analysis for the 3-RPS and delta-like family parallel manipulators. The cylindrical algebraic decomposition method and Gröbner based computations are used to model the workspace in $3T \& 2R1T$ projection spaces, where the orientation of the mobile platform is represented using quaternions. A certified three dimensional plotting is proposed to study the shape of the workspace for different delta like manipulators. Also, the influence of the design parameters and joint limits on the workspace boundaries for the parallel manipulators are presented.

II.1 Introduction

The workspace can be defined as the volume of space or the complete set of poses which the end-effector of the manipulator can reach. Many researchers published several works on the problem of computing these complete sets for robot kinematics. Based on the early studies [1, 2], several methods for workspace determination have been proposed, but many of them are applicable only for a particular class of robots. The workspace of parallel robots mainly depends on the actuated joint variables, the range of motion of the joints and the mechanical interferences between the bodies of mechanism. There are different techniques based on geometric [3, 4], discretization [5, 7, 8], and algebraic methods [9, 10, 11, 12, 22, 23, 24] which can be used to compute the workspace of parallel robot. The main advantage of the geometric approach is that, it establish the nature of the boundary of the workspace [19]. Also it allows to compute the surface and volume of the workspace while being very efficient in terms of storage space, but when the rotational motion is included, it becomes less efficient. Interval analysis based methods can be used to compute the workspace but the computation time depends on the complexity of the robot and the requested accuracy [8]. Discretization methods are usually less complicated and can easily take into account

all kinematic constraints, but they require more space and computation time for higher resolutions.

The majority of numerical methods which is used to determine the workspace of parallel manipulators includes the discretization of the pose parameters for computing workspace boundaries [7]. There are other approaches, which are based on optimization algorithms [20] for fully serial or parallel manipulators, analytic methods for symmetrical spherical mechanisms [21]. In [74] a method for computing the workspace boundary for manipulators with a general structure is proposed, which uses a branch-and-prune technique to isolate a set of output singularities, and then classifies the points on such set according to whether they correspond to motion impediments in the workspace. A cylindrical algebraic decomposition based method is illustrated in [11, 16], which is used to model the workspace and joint space for the 3 RPS parallel robot.

II.2 Manipulators Under Study

There are three different mechanisms for which the workspace and the joint space analysis are presented in this chapter. The first mechanism under the study is a three degree of freedom parallel mechanism which consists of three identical legs, the different arrangements of these legs give rise to family of delta like robot. Several types of delta-like robot were studied, few of them are Orthoglide [8, 76], Hybridglide, Triaglide [78] and UraNeSX [8]. The second robot is a reconfigurable 3RPS parallel robot with variable mobile platform. The third robot under the study is the 3RRR parallel robot, the NaVARo robot. SIROPA library Provides modeling, analyzing and plotting functions for different manipulators. The function *CreateManipulator()* of SIROPA library in MAPLE software is used to create the manipulator virtually for analyzing the different parameter associated with the manipulator.

II.2.1 Orthoglide Architecture and Kinematics

The *Orthoglide* mechanism is driven by three actuated orthogonal prismatic joints. A simpler virtual model can be defined for the Orthoglide, which consists of three bar links connected by the revolute joints to the tool center point on one side and to the corresponding prismatic joint at another side. Several assembly modes of these robots depends upon the solutions of direct kinematic problem is shown in Fig. II.1. The point \mathbf{P} represents the pose of corresponding robot. However more than one value for the point \mathbf{P} shows the multiple solutions for the DKP. $\mathbf{A}_i\mathbf{B}_i$ is equal to ρ_i , where ρ_i represents the prismatic joint variables whereas \mathbf{P} represents the position vector of the tool center point.

The constraint equations for the Orthoglide are:

$$\begin{aligned}(x - \rho_1)^2 + y^2 + z^2 &= L^2 \\ x^2 + (y - \rho_2)^2 + z^2 &= L^2 \\ x^2 + y^2 + (z - \rho_3)^2 &= L^2\end{aligned}\tag{II.1}$$

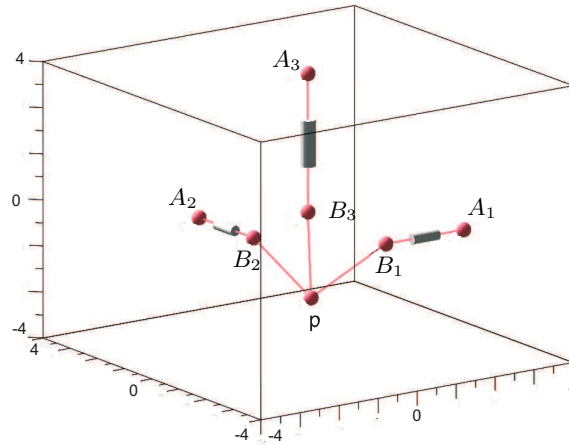


Figure II.1 – Configuration plot for Orthoglide robot

II.2.2 Hybridglide Architecture and Kinematics

The *Hybridglide* mechanism consists of three actuated prismatic joints, in which two actuators are placed parallel and third one perpendicular to others two. Also the three bar links connected by spherical joints to the tool center point on one side and to the corresponding prismatic joint at another side. Several assembly modes of these robots depends upon the solutions of direct kinematic problem is shown in Fig. II.2.

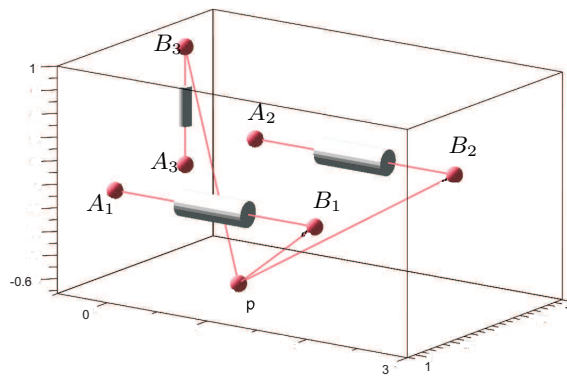


Figure II.2 – Configuration plot for Hybridglide robot

The constraint equations for the Hybridglide are:

$$\begin{aligned}
 (x - 1)^2 + (y - \rho_1)^2 + z^2 &= L^2 \\
 (x + 1)^2 + (y - \rho_2)^2 + z^2 &= L^2 \\
 x^2 + y^2 + (z - \rho_3)^2 &= L^2
 \end{aligned}
 \tag{II.2}$$

II.2.3 Triaglide Architecture and Kinematics

The *Triaglide* manipulator is driven by three actuated prismatic joints, in which all the three actuators parallel to each other and placed in the same plane. The architecture of the Triaglide is shown in Fig. II.3.

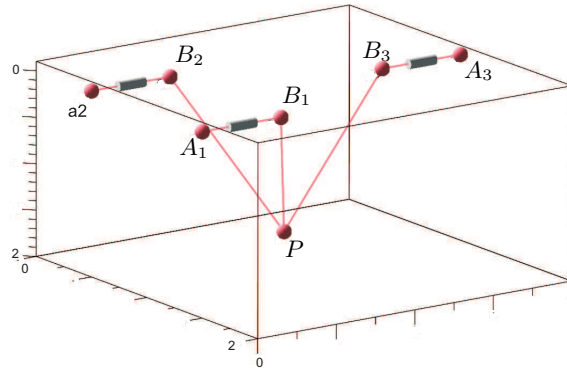


Figure II.3 – Configuration plot for Triaglide robot

The constraint equations for the Triaglide are:

$$\begin{aligned} (x - 1)^2 + (y - \rho_1)^2 + z^2 &= L^2 \\ (x + 1)^2 + (y - \rho_2)^2 + z^2 &= L^2 \\ x^2 + (y - \rho_3)^2 + z^2 &= L^2 \end{aligned} \quad (\text{II.3})$$

II.2.4 UraneSX Architecture and Kinematics

The *UraneSX* is similar to triaglide, but instead of three actuators in the same plane, they are placed in different planes. The architecture of the UraneSX is shown in Fig. II.4.

The constraint equations for the UraneSX are:

$$\begin{aligned} (x - 1)^2 + y^2 + (z - \rho_1)^2 &= L^2 \\ (x + 1/2)^2 + (y - \sqrt{3}/2)^2 + (z - \rho_2)^2 &= L^2 \\ (x + 1/2)^2 + (y + \sqrt{3}/2)^2 + (z - \rho_3)^2 &= L^2 \end{aligned} \quad (\text{II.4})$$

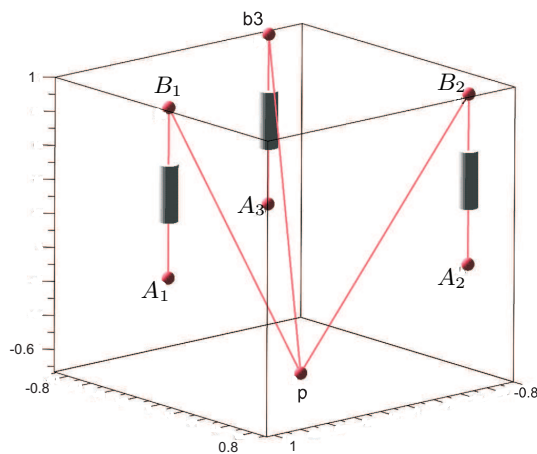


Figure II.4 – Configuration plot for UraneSX robot

II.2.5 Reconfigurable 3-RPS Parallel Robot

The robot under study is the 3-RPS parallel robot with three degrees of freedom and has been studied by many researchers [79, 80]. It is the assembly of two equilateral triangles (the base, moving platform) by three identical RPS legs where R is a revolute passive joint, P an actuated prismatic joint and S a passive spherical joint. Thus, the revolute joint is connected to the fixed base and the spherical joint to the mobile platform.

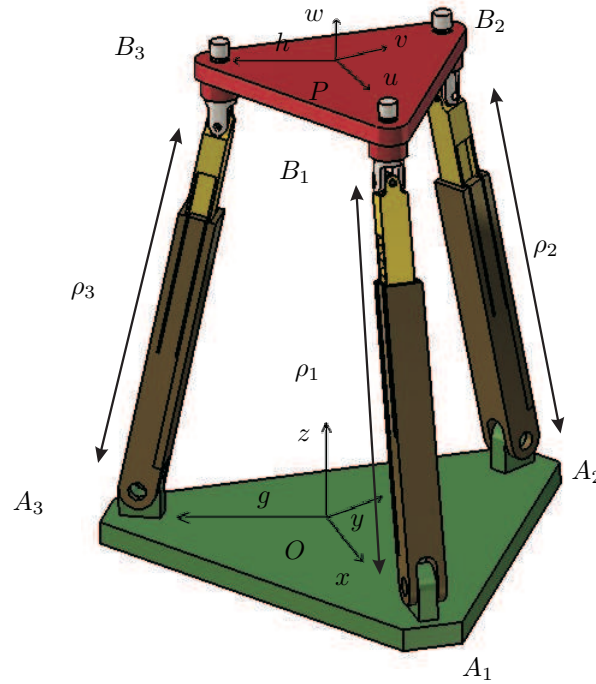


Figure II.5 – 3-RPS parallel robot

Considering the 3-RPS parallel manipulator, as shown in Fig. II.5, the fixed base consists of an equilateral triangle with vertices A_1 , A_2 and A_3 , and circumradius g . The moving platform is another equilateral triangle with vertices B_1 , B_2 and B_3 and varying circumradius h , and circumcenter P . The two design parameters g and h are positive numbers. Connecting each of the vertices' pairs A_i , B_i ($i = 1, 2, 3$) by a limb, a rotational joint lies at A_i and a spherical joint lies at B_i . ρ_i denotes the length of each limb and the adjustment is done through an actuated prismatic joint. Thus, there are five parameters, namely g , h , ρ_1 , ρ_2 and ρ_3 . The g and h parameters ($\mathbf{h}/\mathbf{g}=\mathbf{k}$) determine the design of the manipulator (shown in fig. II.6) whereas the joint parameters ρ_1 , ρ_2 and ρ_3 determine the motion of the robot. Spatial rotations in three dimensions can be parametrized using Euler angles [81], unit quaternions [82] or dual quaternions [79]. The quaternion representation is used for modeling the orientation as quaternions do not suffer from singularities as Euler angles do. Moreover, to transform the trigonometric equations to algebraic equations, we may either introduce the singularity of the transformation $t = \tan(\alpha/2)$ or replace the angle α by two parameters \cos_α and \sin_α with $\cos_\alpha^2 + \sin_\alpha^2 = 1$. In addition, it is easier to represent workspace sections with the quaternions than the dual quaternions.

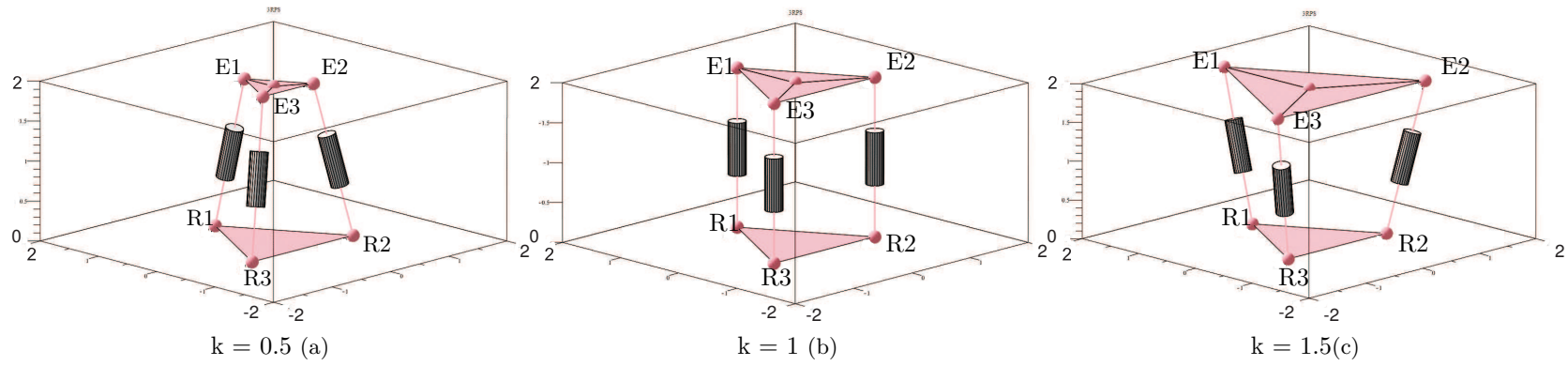


Figure II.6 – Virtual model of 3-RPS parallel manipulator with different design parameter k , $k = 0.5$ (a), $k = 1$ (b), $k = 1.5$ (c)

A quaternion \mathbf{q} is defined by

$$\mathbf{q} = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k} \quad (\text{II.5})$$

The quaternion rotation matrix for the parallel robot is then

$$\mathbf{Q} = \begin{bmatrix} 2q_1^2 + 2q_2^2 - 1 & -2q_1q_4 + 2q_2q_3 & 2q_1q_3 + 2q_2q_4 \\ 2q_1q_4 + 2q_2q_3 & 2q_1^2 + 2q_3^2 - 1 & -2q_1q_2 + 2q_3q_4 \\ -2q_1q_3 + 2q_2q_4 & 2q_1q_2 + 2q_3q_4 & 2q_1^2 + 2q_4^2 - 1 \end{bmatrix} \quad (\text{II.6})$$

with $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$. The transformation from the moving frame to the fixed frame can be described by a position vector $\mathbf{p} = OP$ and a 3×3 rotation matrix \mathbf{R} . Let \mathbf{u} , \mathbf{v} and \mathbf{w} be the three unit vectors defined along the axes of the moving frame, then the rotation matrix can be expressed in terms of the coordinates of \mathbf{u} , \mathbf{v} and \mathbf{w} as:

$$\mathbf{R} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \quad (\text{II.7})$$

The vertices of the base triangle and mobile platform triangle are

$$\mathbf{A}_1 = \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{A}_2 = \begin{bmatrix} -g/2 \\ g\sqrt{3}/2 \\ 0 \end{bmatrix} \quad \mathbf{A}_3 = \begin{bmatrix} -g/2 \\ -g\sqrt{3}/2 \\ 0 \end{bmatrix} \quad (\text{II.8})$$

$$\mathbf{b}_1 = \begin{bmatrix} h \\ 0 \\ 0 \end{bmatrix} \quad \mathbf{b}_2 = \begin{bmatrix} -h/2 \\ h\sqrt{3}/2 \\ 0 \end{bmatrix} \quad \mathbf{b}_3 = \begin{bmatrix} -h/2 \\ -h\sqrt{3}/2 \\ 0 \end{bmatrix} \quad (\text{II.9})$$

The coordinates of \mathbf{b}_i with respect to fixed frame reference are obtained by $\mathbf{B}_i = \mathbf{P} + \mathbf{R}\mathbf{b}_i$ for $i = 1, 2, 3$. Also the coordinates of the centre of the mobile platform in the fixed reference is $\mathbf{P} = [x \ y \ z]^T$. The distance constraints yields:

$$\|\mathbf{A}_i - \mathbf{B}_i\| = \rho_i^2 \quad \text{with } i = 1, 2, 3 \quad (\text{II.10})$$

As A_i are revolute joints, the motion of the B_i are constrained in planes. This leads to the three constraint equations:

$$u_y h + y = 0 \quad (\text{II.11})$$

$$y - u_y h/2 + \sqrt{3}v_y h/2 + \sqrt{3}x - \sqrt{3}u_x h/2 + 3v_x h/2 = 0 \quad (\text{II.12})$$

$$y - u_y h/2 - \sqrt{3}v_y h/2 - \sqrt{3}x + \sqrt{3}u_x h/2 + 3v_x h/2 = 0 \quad (\text{II.13})$$

Solving with respect to x and y we get:

$$y = -hu_y \quad (\text{II.14})$$

$$x = h \left(\sqrt{3}u_x - \sqrt{3}v_y - 3u_y + 3v_x \right) \sqrt{3}/6 \quad (\text{II.15})$$

In Equations II.10, II.12, II.13, we substitute x, y using relations II.14 and II.15, and $\mathbf{u}, \mathbf{v}, \mathbf{w}$ by quaternion expressions using II.6. Then (II.12) and (II.13) become $q_1q_4 = 0$. Thus, we have either $q_1 = 0$ or $q_4 = 0$. This property is associated with the notion of operation mode. To obtain the algebraic equations, we replace $\sqrt{3}$ by the variable S_3 and add the equation $S_3^2 - 3 = 0$ and the constraint $s_3 > 0$.

II.2.6 NaVARo 3RRR Planar Parallel Robot

The NaVARo robot was introduced in [86] and illustrated in Fig. IV.12. From the classical 3-RRR parallel robot, a parallelogram is added to each leg and connected to the first limb. A virtual model of NaVARo robot is shown in Fig. IV.13 which is used for analyzing the different aspects associated with the robot under study. Virtually, the actuator positions can change from the revolute joint located on A_i or B_i . This feature exists thanks to a clever transmission system made with two clutches. The NaVARo robot admits eight actuation modes if we assume that only one clutch can be activated in the same time per leg.



Figure II.7 – The NaVARo robot

The dimensions were chosen so that the forward kinematics of the 3-RPR robot associated can be solved analytically. Thus, the points A_i and C_i form an equilateral triangle, respectively, and their sides has the dimension 0.7 m and 0.35 m, respectively. The lengths of the links A_iB_i and B_iC_i are 0.21 m for $i = 1, 2, 3$. Due to the parallelograms, the minimum and maximum distance between the points A_i and C_i is 0.039 m and 0.08 m, respectively.

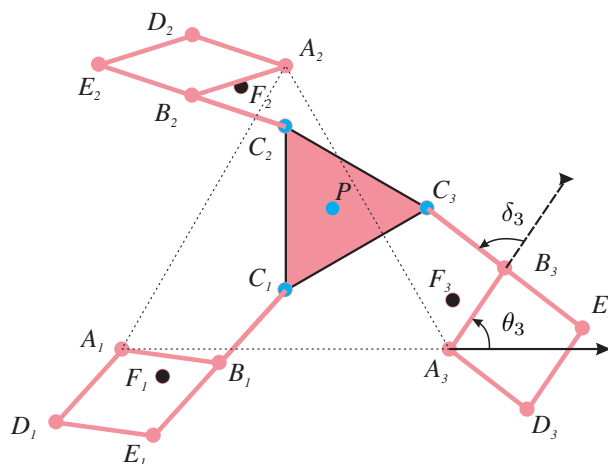


Figure II.8 – The virtual model of the NaVARo robot

II.3 Workspace Analysis of a Delta-like Family Robot

The workspace analysis allows to characterize of the workspace regions where the number of real solutions for the inverse kinematics is constant. A CAD algorithm is used to compute the workspace of the robot in the projection space (x, y, z) with some joint constraints taken in account.

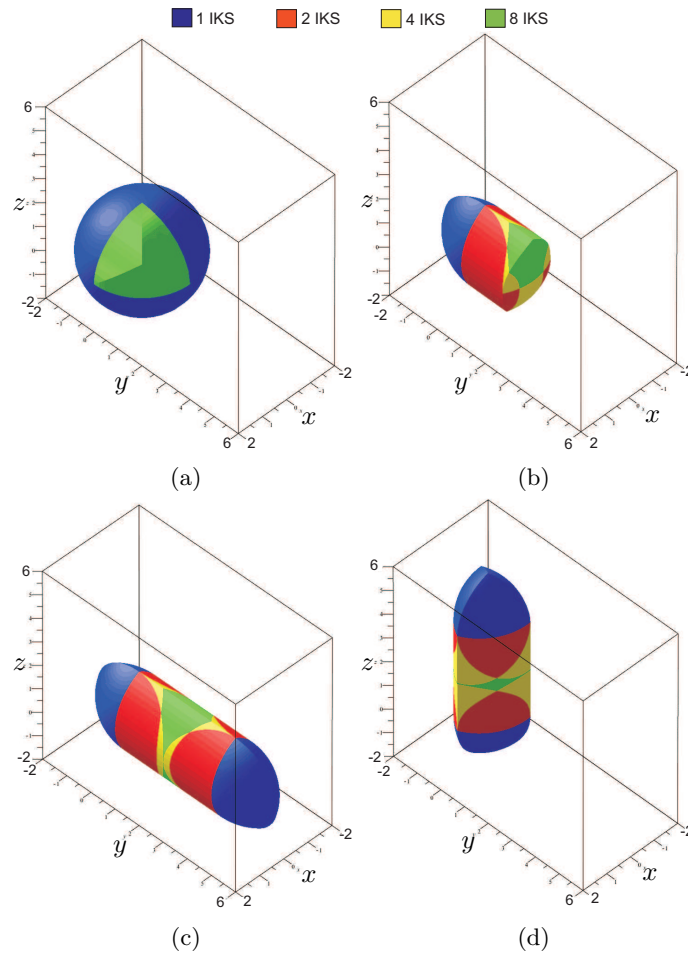


Figure II.9 – Workspace plot for Orthoglide (a), Hybridglide (b), Triaglide (c) and UraneSX (d) robot

The three main steps involved in the analysis are [16, 52, 56]:

- Computation of a subset of the joint space (resp. workspace) where the number of solutions changes: the *Discriminant Variety*.
- Description of the complementary of the discriminant variety in connected cells: the Generic *Cylindrical Algebraic Decomposition* (CAD).
- Connecting the cells belonging to the same connected component in the counterpart of the discriminant variety: *interval comparisons*.

Table II.1 shows the number of cells corresponding to the number of solutions in the

Table II.1 – Definition of the Workspace with CAD

WORKSPACE						
Number of cells						
Number of solutions	0	1	2	4	8	Total
Orthoglide	28782	1196	0	0	130	30108
Hybridglide	93292	4484	7228	4196	1164	110364
Triaglide	27708	384	464	420	400	29376
UraneSX	9918	236	36	0	0	10190

jointspace, which is the outcome of cell decomposition. The different shapes of workspace for the delta-like robots is shown in Fig. II.9, where blue, red, yellow and green regions correspond to the one, two, four and eight number of solutions for the IKP. A comparative study is done on the workspace of the family of delta-like manipulator and the results are shown in Fig. II.9. All the workspace are plotted in the rectangular box, where $\mathbf{x} \in [-2, 2]$, $\mathbf{y} \in [-2, 6]$ and $\mathbf{z} \in [-2, 6]$, so that the shapes of these workspace can be compared. From the Fig. II.9 it can be intuited that the Triaglide will be good selection, if the task space is more in horizontal plane, whereas the Orthoglide is good for the three dimensional task space.

II.4 Joint space Analysis of a Delta-like Family Robot

The Jointspace analysis predicts the feasible and non-feasible combinations of the prismatic joint variables which are essential for the parallel robot control. The Jointspace analysis is done using CAD. Table II.2 shows the number of cells corresponding to the number of solutions in the jointspace, which is the outcome of cell decomposition. Also these cells are plotted to represent the jointspace which is shown in Figure IV.16, where red region corresponds to two solutions for the DKP. The following joint constraints are taken into account while computing the joint space of the manipulators.

$$0 < \rho_1 < 2L \quad 0 < \rho_2 < 2L \quad 0 < \rho_3 < 2L \quad (\text{II.16})$$

Table II.2 – Definition of the Jointspace with CAD

JOINTSPACE			
Number of cells			
Number of solutions	0	2	Total
Orthoglide	10509	4160	14669
Hybridglide	8917	3041	11958
Triaglide	5375	426	5801
UraneSX	50598	4006	54604

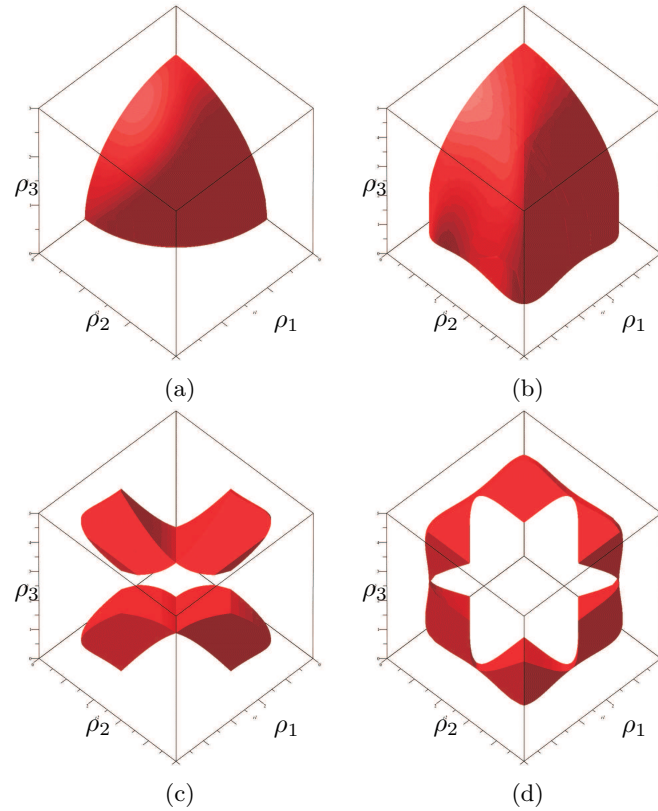


Figure II.10 – Jointspace plot for Orthoglide (a), Hybridglide (b), Triaglide (c) and UraeSX (d) robot

II.5 Workspace Analysis of the 3RPS Parallel robot

The workspace of the robot is a cylinder in the projection space (z, q_2, q_3) if there are no joint limits on the actuated joints.

The workspace analysis can be done by dividing it into a set of aspects. The notion of aspect, previously defined in [59] for serial robots and in [17] for parallel robot with one operation mode can be extended for a parallel robot with several operation modes such that:

- $WA_i^j \subset W_{OM^j}$
- WA_i^j is connected
- $\forall X \in WA_i^j, \det(\mathbf{A}) \neq 0$ and $\det(\mathbf{B}) \neq 0$

In other words, an aspect WA_i^j is the largest connected region without any singularity of the OM^j .

The analysis of the workspace is done in the projection space (z, q_2, q_3) . We found out four aspects as shown in Figure III.9. However, no further analysis is done to prove this feature in the four dimension space. As there are several solutions for the DKP in the same aspect, non-singular assembly mode trajectories are possible [79].

The cylindrical algebraic decomposition cannot be used alone to compute the workspace in the projection space of z, q_2, q_3 because the projection of q_1 or q_4 induces spurious surfaces

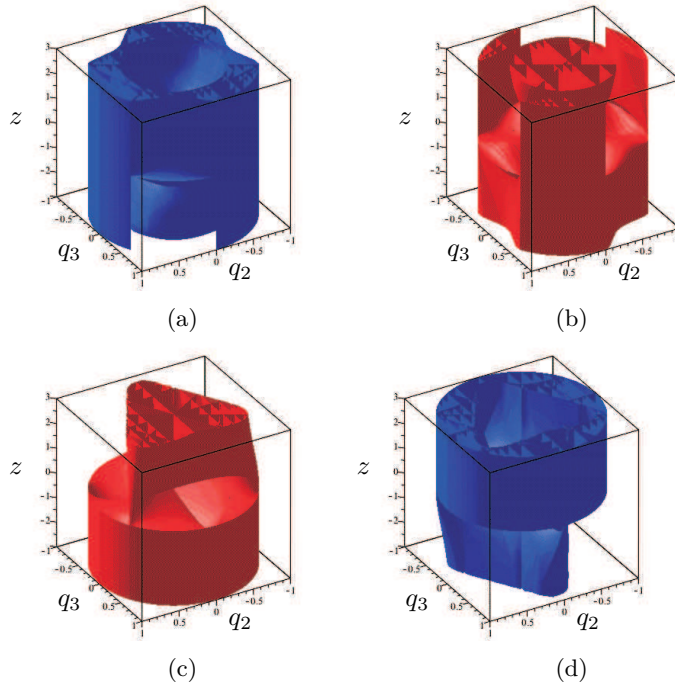


Figure II.11 – Aspects for OM^1 with $\det(\mathbf{A}) < 0$ (a) and $\det(\mathbf{A}) > 0$ (b) and aspects for OM^2 with $\det(\mathbf{A}) < 0$ (c) and $\det(\mathbf{A}) > 0$ (d)

in the space (z, q_2, q_3) . We thus have to solve the inverse kinematic problem for each test point with the constraint $q_4 > 0$ for OM^1 and $q_1 > 0$ for OM^2 .

II.5.1 Operation Modes

Notion of operation mode (OM) was introduced in [58] to explain the behavior of the DYMO robot. An operation mode is associated with a specific type of motion. For the DYMO robot, there are five operation modes: translational, rotational, planar (two types) and mixed motions. In the workspace W , for each operation mode, W_{OM^j} is defined such that

- $W_{OM^j} \subset W$
- $\forall X \in W_{OM^j}$, OM is constant

For a parallel robot with several operating modes, the pose can be defined by fixing the same number of parameters as the degree of freedom of the mobile platform. Given an operation mode OM^j , if we have a single inverse kinematic solution, we can define an application that maps \mathbf{X} onto \mathbf{q} :

$$g_j(\mathbf{X}) = \mathbf{q} \quad (\text{II.17})$$

Then, the images in W_{OM^j} of a posture \mathbf{q} in the joint space Q are defined by:

$$g_j^{-1}(\mathbf{q}) = (\mathbf{X} / (\mathbf{X}, \mathbf{q}) \in OM^j) \quad (\text{II.18})$$

where g_j^{-1} stands to be the direct kinematic problem for the operation mode j .

II.5.2 Cylindrical Algebraic Decomposition

The workspace (resp. Joint space) analysis classifies the number of solutions of the parametric system associated with the Inverse (resp. Direct) Kinematic Problem (IKP). This method was introduced for parallel robots in [12]. We will recall here the main steps of the computation and the new step for a robot with several operation modes. For such operations, both workspace as well as joint space is decomposed into cells C_1, \dots, C_k , such that:

- C_i is an open connected subset of the workspace;
- for all pose values in C_i , the direct (resp. inverse) kinematics problem has a constant number of solutions;
- C_i is maximal in the sense that if C_i is contained in a set E , then E does not satisfy the first or the second condition.
- The C_i are disjoint and their complementary is a set of null measure.

The three main steps involved in the analysis are:

- Computation of a subset of the joint space (resp. workspace) where the number of solutions changes: the *Discriminant Variety* [12].
- Description of the complementary of the discriminant variety in connected cells: the *Generic Cylindrical Algebraic Decomposition*.
- Connecting the cells belonging to the same connected component in the counterpart of the discriminant variety: *interval comparisons*.

From a general point of view, the discriminant variety is defined for any system of polynomial equations and inequalities. The union of the Discriminant variety and of the cells thus define a partition of the considered space. Let p_1, \dots, p_m and q_1, \dots, q_l be polynomials with rational coefficients depending on the unknowns X_1, \dots, X_n , and on the parameters U_1, \dots, U_d . Let us consider the constructible set:

$$\mathcal{C} = \left\{ \begin{array}{l} \mathbf{v} \in \mathbb{C}^{n+d}, p_1(\mathbf{v}) = 0, \dots, p_m(\mathbf{v}) = 0, \\ q_1(\mathbf{v}) \neq 0, \dots, q_l(\mathbf{v}) \neq 0 \end{array} \right\} \quad (\text{II.19})$$

If we assume that \mathcal{C} is a finite number of points for almost all the parameter values, a discriminant variety V_D of \mathcal{C} is a variety in the parameter space \mathbb{C}^d such that, over each connected open set \mathcal{U} satisfying $\mathcal{U} \cap V_D = \emptyset$, \mathcal{C} defines an analytic covering. In particular, the number of points of \mathcal{C} over any point of \mathcal{U} is constant.

Let us now consider the following semi-algebraic set:

$$\mathcal{S} = \left\{ \begin{array}{l} \mathbf{v} \in \mathbb{C}^{n+d}, p_1(\mathbf{v}) = 0, \dots, \\ p_m(\mathbf{v}) = 0, q_1(\mathbf{v}) \geq 0, \dots, q_l(\mathbf{v}) \geq 0 \end{array} \right\} \quad (\text{II.20})$$

If we assume that \mathcal{S} has a finite number of solutions over at least one real point that does not belong to V_D , then $V_D \cap \mathbb{R}^d$ can be viewed as a real discriminant variety of $\mathcal{S} \cap \mathbb{R}^{n+d}$, with the same property: over each open set $\mathcal{U} \subset \mathbb{R}^d$ such that $\mathcal{U} \cap V_D \cap \mathbb{R}^d = \emptyset$, \mathcal{C} defines

an analytic covering. In particular the number of real points of \mathcal{S} over any point of \mathcal{U} is constant. Discriminant varieties can be computed using basic and well known tools from computer algebra such as Gröbner bases [83] and a full package computing such objects in a general framework is available in Maple software through the RootFinding[Parametric] package.

The CAD implemented in the SIROPA library has been used to compute the aspects into a set of cells where algebraic equations define its boundaries and a sample point in each one [12] for the 2PRR-RPR parallel robot. For example, the CAD can provide a formal decomposition of the joint space in cells where the polynomials $\det(\mathbf{A})$ and $\det(\mathbf{B})$ have a constant sign and the number for the DKP is constant [73].

II.5.3 Workspace: 3T Projection Space for OM^1 and OM^2

The 3-RPS parallel robot is three degrees of freedom parallel robot with two rotational [2R] and one translational [1T] motions, but still there exist two other translational motions in the x and y direction. These motions are termed as parasitic motions as they depend on other existing rotational motions. A complete workspace analysis of this mechanism is possible, only when the analysis is done in both 2R1T $[p_z, q_2, q_3]$ and 3T $[p_x, p_y, p_z]$ projection spaces for OM_1 and OM_2 . The variation in the workspace boundaries due to the design parameter k for both the operation modes is shown in Fig. II.12. Blue and red regions corresponds to the four number of solutions for the IKP for $\det(\mathbf{A}) > 0$ and $\det(\mathbf{A}) < 0$, respectively. These analyses can be useful for the researchers or engineers to select the optimum value for the design parameter such that the parasitic motions can be limited to specific values.

From Fig. II.12, it can be depicted that as the value of k increases, singularity becomes more complex and there exists a larger area without singularities for smaller values of k .

II.6 Joint space Analysis of the 3RPS Parallel robot

The joint space analysis allows the characterization of the regions where the number of real solutions for the direct kinematic model is constant. Using CAD, we can do this study on sections of the joint space. The calculation for the full joint space is possible, but the number of cells obtained is too large for the display capabilities of Maple. Without taking into account the notion of operation mode, Figure II.13 depicts the regions with 4, 8, 12 or 16 solutions for the DKP. The maximum number for the DKP of each operation mode is 8.

Figures II.14 and II.17 show several slices of the joint space for OM^1 and OM^2 , where the DKP admits four and eight real solutions in the blue and red region respectively.

Cuspidal configurations are associated with second-order degeneracies that appear for triply coalesced configurations. These configurations play an important role in the path planning because they are directly linked to the non-singular assembly mode changing trajectories [63, 70, 68, 14]. A state of the art for the computation of the cusp points is given in [13]. Here we make use of the Jacobian criterion on the singular locus to extract the points of multiplicity greater than or equal to three. Figures II.14 and II.17 show that the number of cusp points is changing according to the value of ρ_1 . Figure II.15 depicts the locus of the

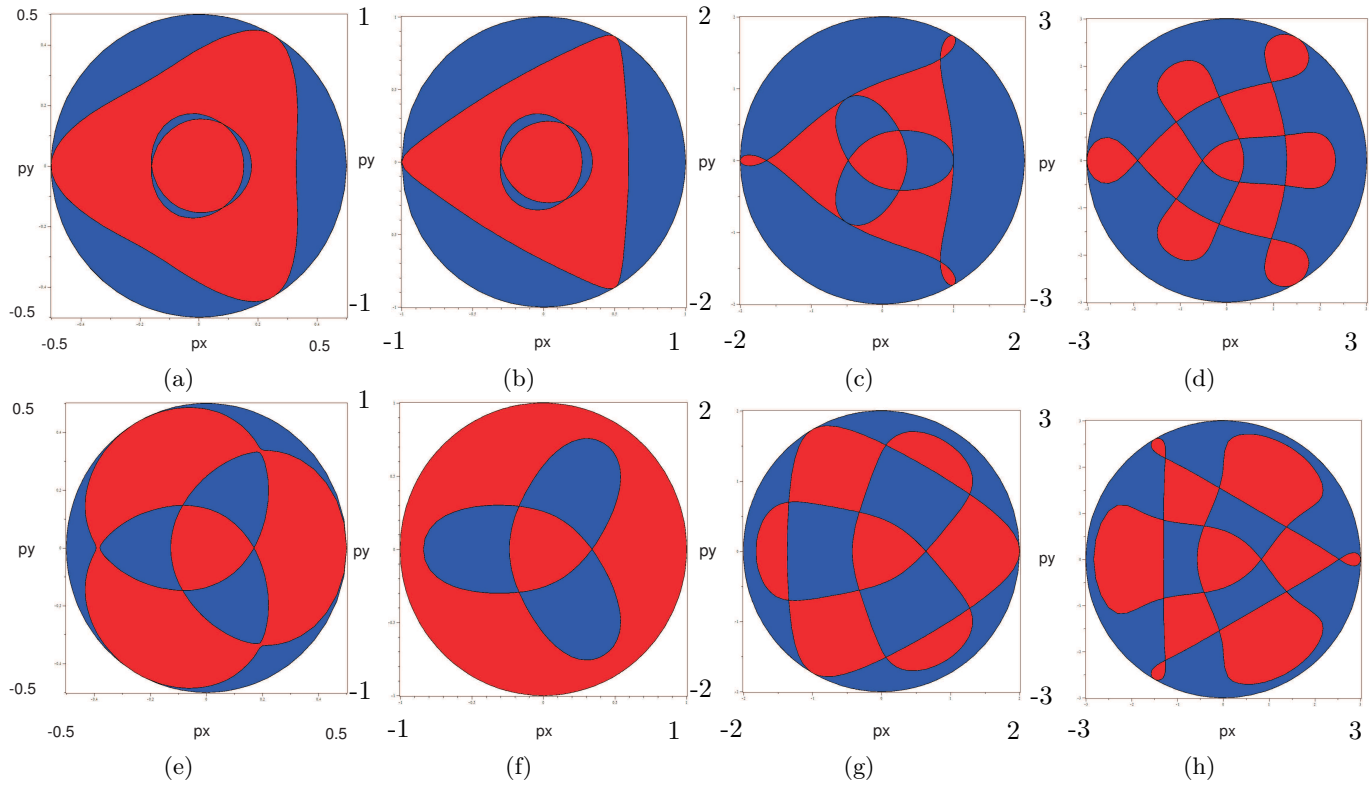


Figure II.12 – Slice of workspace for OM^1 with different design parameter and $p_z = 2$, $k = 0.5$ (a), $k = 1$ (b), $k = 2$ (c) and $k = 3$ (d) for OM^2 $k = 0.5$ (e), $k = 1$ (f), $k = 2$ (g) and $k = 3$ (h). Blue and red regions corresponds to the four number of solutions for the IKP for $\det(\mathbf{A}) > 0$ and $\det(\mathbf{A}) < 0$, respectively.

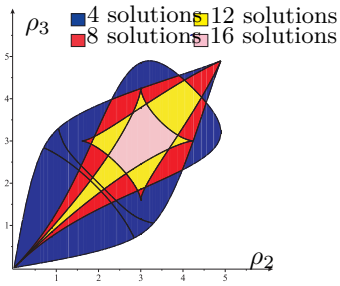


Figure II.13 – Slice of the joint space for $\rho_1 = 3$ and the number of solution for DKP for $k = 1$

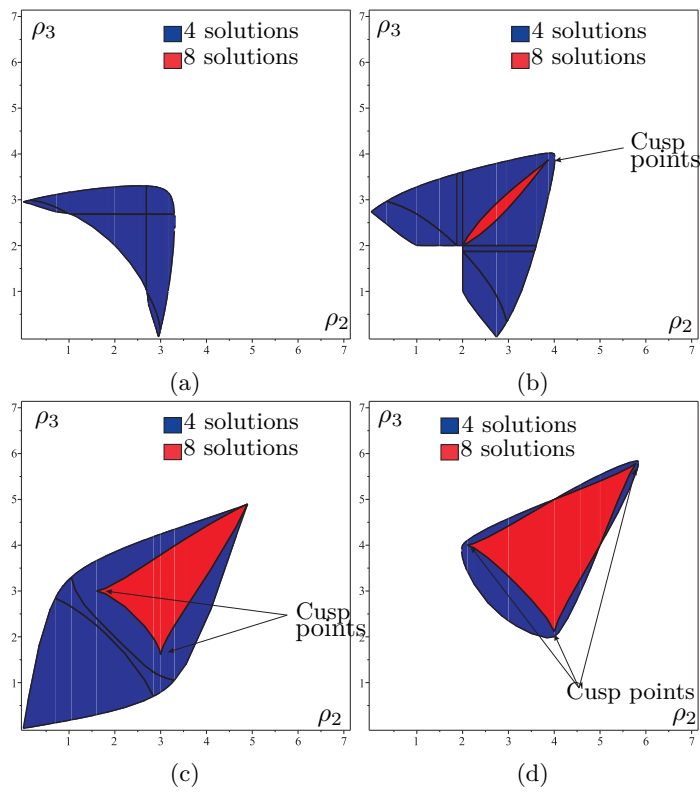


Figure II.14 – Slice of the joint space for OM^1 for $\rho_1 = 1$ (a), $\rho_1 = 2$ (b), $\rho_1 = 3$ (c) and $\rho_1 = 4$ (d) and the number of solution for DKP for $k = 1$

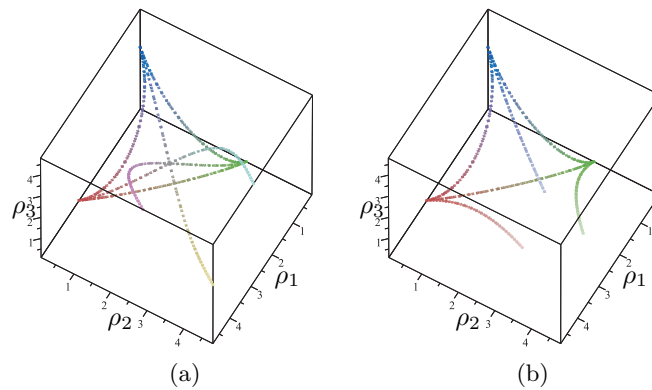


Figure II.15 – Cusp points for OM^1 (a) and OM^2 (b) for $k = 1$

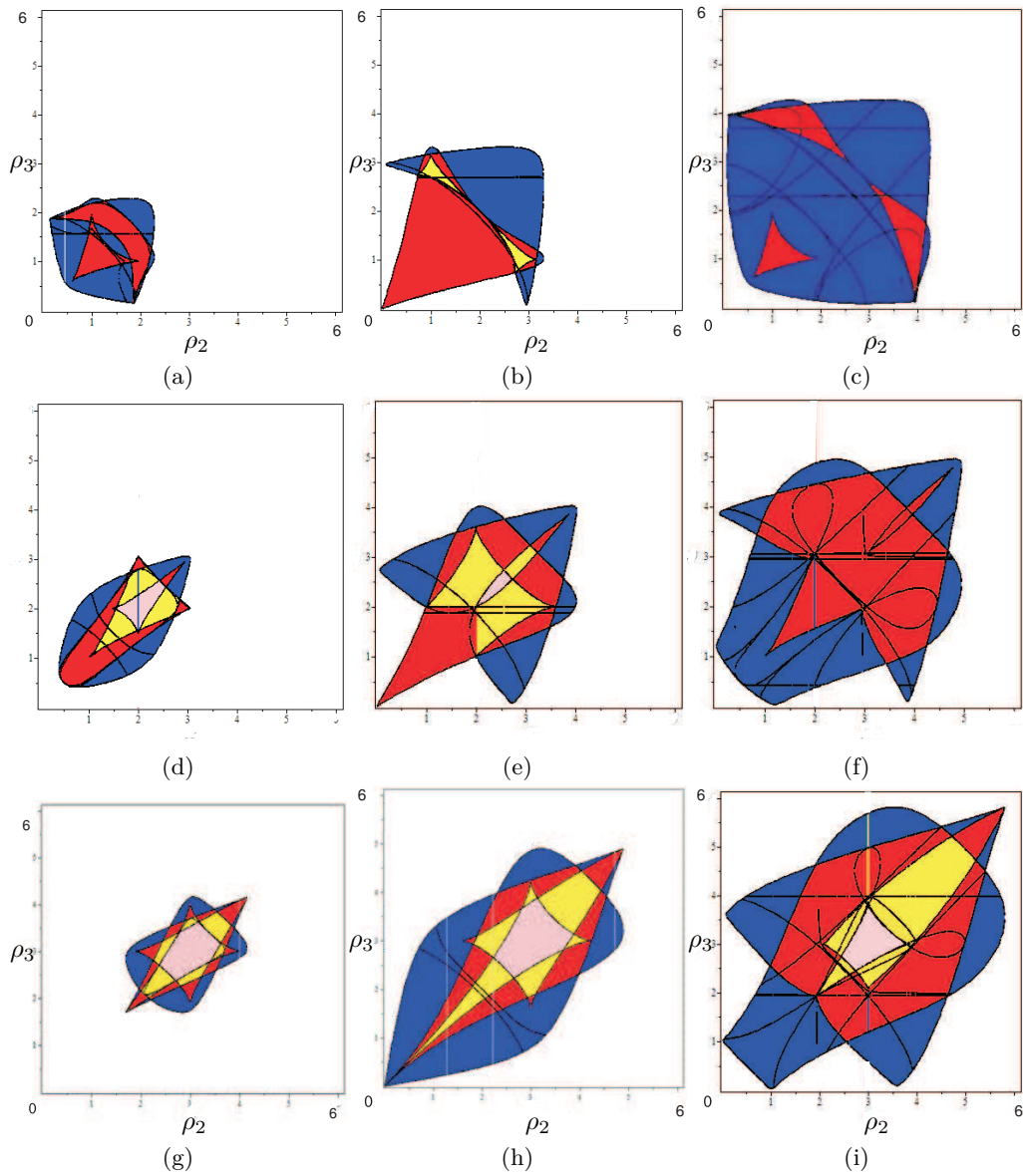


Figure II.16 – Slice of the joint space for $\rho_1 = 1, k = 0.5$ (a), $\rho_1 = 1, k = 1$ (b), $\rho_1 = 1, k = 1.5$ (c), $\rho_1 = 2, k = 0.5$ (d), $\rho_1 = 2, k = 1$ (e), $\rho_1 = 2, k = 1.5$ (f), $\rho_1 = 3, k = 0.5$ (g), $\rho_1 = 3, k = 1$ (h) $\rho_1 = 3, k = 1.5$ (i), where the DKP admits four, eight, twelve and sixteen real solutions in the blue, red, yellow and pink region, respectively.

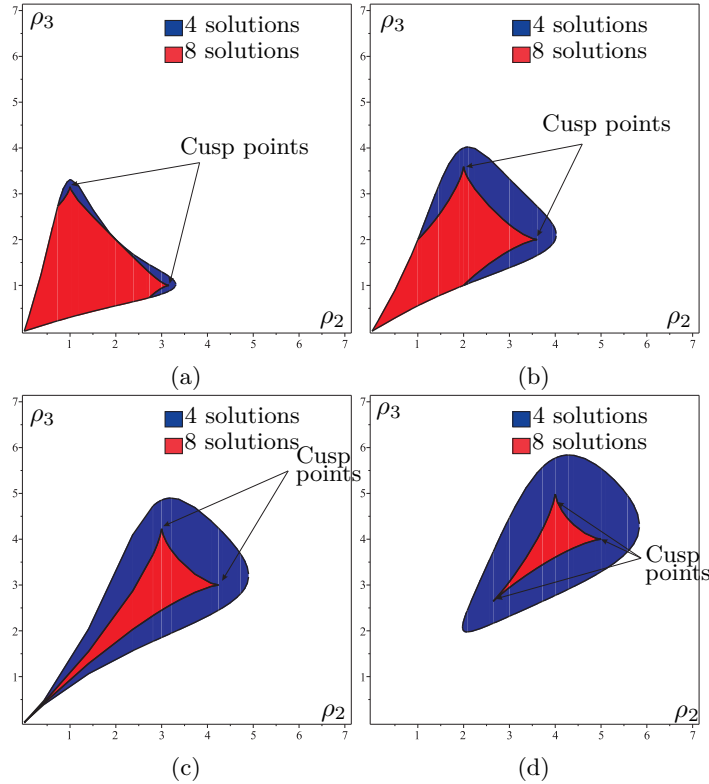


Figure II.17 – Slice of the joint space for OM^2 for $\rho_1 = 1$ (a), $\rho_1 = 2$ (b), $\rho_1 = 3$ (c) and $\rho_1 = 4$ (d) and the number of solution for DKP

cusps in the joint space. These curves are obtained as the intersection of 11 surfaces for OM^1 and 8 surfaces for OM^2 . However, for the 3-RPS parallel robot is very difficult to remove the multiplicity greater than two by using the saturation method as introduced in [48].

The variation in the joint space boundaries due to the design parameter k is shown in Figure II.16. The slice of joint space is computed for the three different values of ρ_1 and k . For $\rho_2 = 2$, as the value of k increases the cusps corresponding to the region with 16 numbers of solution for DKP disappears (in Fig. II.16(d), II.16(e) & II.16(f)). The increase in the area of the joint space with an increase in the value of k , shows the more feasible actuation sets compared with the configuration with lesser value of k .

II.7 Conclusions

This work reports the variations in the workspace and the joint space with respect to the design parameter 'k' of the 3-RPS parallel manipulator. The cylindrical algebraic decomposition method and Gröbner based computations are used to model the workspace and joint space with the parallel singularities in 3T & 2R1T projection spaces, where the orientation of the mobile platform is represented using quaternions. Depending on the design parameter 'k', three different configurations of the 3-RPS parallel manipulator are analyzed. A comparative study on the workspace of different delta-like robots gives the idea about shape

of the workspace, which further plays an important role in the selection of the manipulator for the specific task or for the trajectory planning.

III

Aspects and Uniqueness Domains

One of the highly addressed problem associated with the parallel manipulators is the parallel singularities. Singularities of a robotic manipulator are important feature that essentially influence its capabilities. Mathematically, a singular configuration may be defined as rank deficiency of the Jacobian describing the differential mapping from the jointspace to the workspace and vice versa. This chapter presents the singularity analysis for the 3-RPS parallel robot and the delta like family robot. The Gröbner based elimination method is used to compute the projection of the singularities in the Cartesian space and the joint space. The descriptive analysis of a family of delta like robots is presented by using algebraic tools to induce the an estimation about the complexity in representing the singularities in the workspace and the joint space. The generalized notions of the uniqueness domains is also presented for the parallel robot with several operation modes. The effect of joint limits on these singularity surfaces are also presented in the later sections of this chapter.

III.1 Introduction

When designing a robot, the last step is the trajectory planning. The task of the robot is generally defined in the workspace whereas the control loop depends on the joint space parameters. While defining the home pose of the robot, the Cartesian pose and the Joint values of the actuators are known. If the trajectory planning is done in the workspace by analyzing only the determinant of the Jacobian, we can reach a Cartesian pose different from the home pose but with the same joint value. This feature is called a non-singular assembly mode changing trajectory and stands only for the parallel robot.

For such robots, the inverse and direct kinematic problem (DKP) can have several solutions. To cope up with this problem, the notion of aspects was introduced for the serial robot in [59] and for the parallel robot in [17] and [47]. For the serial robots, the aspects are defined as the maximal singularity-free sets in the joint space whereas in case of parallel robots, the aspects are defined as the maximal singularity-free sets in the workspace or the cross-product of the joint space by the workspace. However, there exists robots, referred as cuspidal robots, which are able to change the inverse kinematic solution without passing

through a singularity for serial robots or direct kinematic solution without passing through a singularity for parallel robots [61, 77, 14, 79, 68]. The uniqueness domains are the connected subsets of the aspects induced by the *characteristic surface*. The CAD algorithm is used to study the workspace and joint space, and a Gröbner based elimination process is used to compute the parallel and serial singularities of the manipulator.

III.2 Singularities: Delta-Like Family Robot

Singularities of a robotic manipulator are important feature that essentially influence its capabilities. Mathematically, a singular configuration may be defined as rank deficiency of the Jacobian describing the differential mapping from the jointspace to the workspace and vice versa. Differentiating the constraints equations with respect to time leads to the velocity model:

$$\mathbf{A}\mathbf{t} + \mathbf{B}\dot{\mathbf{q}} = 0 \quad (\text{III.1})$$

where \mathbf{A} and \mathbf{B} are the parallel and serial Jacobian matrices, respectively, \mathbf{t} is the velocity of P and $\dot{\mathbf{q}}$ joint velocities. The parallel singularities occur whenever $\det(\mathbf{A}) = 0$. The serial singularities occur whenever $\det(\mathbf{B}) = 0$. From the constraint equations Eq. II.1, II.2, II.3 and II.4, the inverse kinematic equations can be derived as:

$$\begin{aligned} \rho_1 &= \gamma_1(\mathbf{X}) \\ \rho_2 &= \gamma_2(\mathbf{X}) \\ \rho_3 &= \gamma_3(\mathbf{X}) \end{aligned} \quad (\text{III.2})$$

For the family of delta-like robots, it is more convenient to compute the inverse jacobian, which can be derived by differentiating Eq. II.1, II.2, II.3 and II.4 with respect to \mathbf{x} , \mathbf{y} and \mathbf{z}

$$A = \begin{pmatrix} \frac{\partial \gamma_1}{\partial \mathbf{x}} & \frac{\partial \gamma_1}{\partial \mathbf{y}} & \frac{\partial \gamma_1}{\partial \mathbf{z}} \\ \frac{\partial \gamma_2}{\partial \mathbf{x}} & \frac{\partial \gamma_2}{\partial \mathbf{y}} & \frac{\partial \gamma_2}{\partial \mathbf{z}} \\ \frac{\partial \gamma_3}{\partial \mathbf{x}} & \frac{\partial \gamma_3}{\partial \mathbf{y}} & \frac{\partial \gamma_3}{\partial \mathbf{z}} \end{pmatrix} \quad (\text{III.3})$$

III.2.1 Parallel Singularities: Projection in workspace and jointspace

Parallel singularities occur when the determinant of the direct kinematics matrix \mathbf{A} vanishes. The corresponding singular configurations are located inside the workspace. They are particularly undesirable because the manipulator can not resist any force and control is lost. Parallel singularity and its projection curve in workspace and jointspace are calculated using the function *ParallelSingularities()* of SIROPA library in MAPLE software. Below is the MAPLE code for calculating the singularity curves and its projection in jointspace.

```

1 s2 := ParalleSingularities ( robot :: Manipulator )
2 s1_cart := Projection ( s1, robot:-PoseVariables )

```

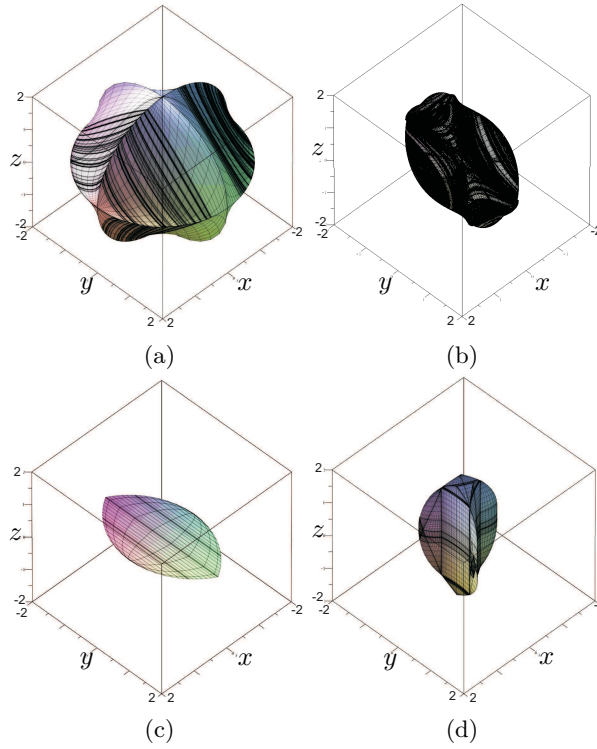


Figure III.1 – Projection of parallel singularity curve of Orthoglide (a), Hybridglide (b), Triaglide (c) and UraneSX (d) in workspace

```
3 s2_art := Projection ( s2, robot:-ArticularVariables )
```

Listing III.1 – Projection of parallel singularity curve of delta-like robot in jointspace

Parallel singularities and their projections in workspace and joint space are computed using a Gröbner based elimination method. This usual way for eliminating variables (see [83]) computes (the algebraic closure of) the projection of the parallel singularities in the workspace.

$$\begin{aligned}
 \det(\mathbf{A})_o &= -8\rho_1\rho_2\rho_3 + 8\rho_1\rho_2z + 8\rho_1\rho_3y + 8\rho_2\rho_3x \\
 \det(\mathbf{A})_h &= -8\rho_1\rho_3x + 8\rho_2\rho_3x - 8\rho_1\rho_3 + 8\rho_1z - 8\rho_2\rho_3 + 8\rho_2z + 16\rho_3y \\
 \det(\mathbf{A})_u &= 4\sqrt{3}(3z - \rho_1 - \rho_2 - \rho_3 + \rho_3x + \rho_2x - 2\rho_1x) + 12\rho_3y - 12\rho_2y \\
 \det(\mathbf{A})_t &= 8\rho_1z + 8\rho_2z - 16\rho_3z
 \end{aligned} \tag{III.4}$$

In the same way, one can compute (the algebraic closure of) the projection of the parallel singularities in the joint space. Both are then defined as the zero set of some system of algebraic equations and we assume that the considered robots are generic enough so that both are hypersurfaces. $\det(\mathbf{A})_o$, $\det(\mathbf{A})_h$, $\det(\mathbf{A})_t$ and $\det(\mathbf{A})_u$ are the parallel singularities of Orthoglide, Hybridglide, Triaglide and UraneSX, respectively. Starting from the constraint equations and the determinant of the Jacobian matrix, we are able to eliminate the joint values. This elimination strategy is more efficient than a cascading elimination by means of resultants which might introduce many more spurious solutions : singular points that are not projections of singular points.

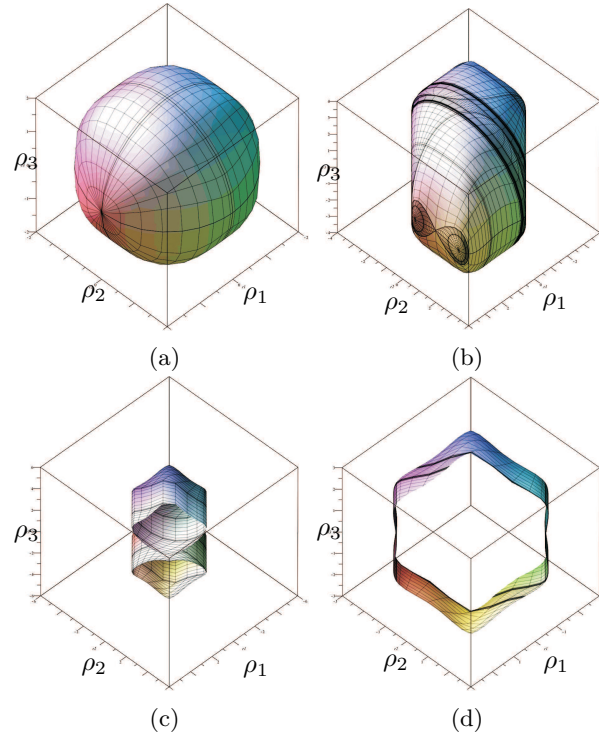


Figure III.2 – Projection of parallel singularity curve of Orthoglide (a), Hybridglide (b), Triaglide (c) and UraneSX (d) in jointspace

Figure III.2 shows the projections of singularity curve s_2 in jointspace and s_2_art is the projection curve. And s_2_art is the projection curve in workspace as shown in Figure III.1.

III.2.2 Serial Singularities: Projection in workspace and jointspace

Serial singularities occur when the determinant of the inverse kinematics matrix \mathbf{B} vanishes. When the manipulator is in such a singularity, there is a direction along which no Cartesian velocity can be produced. Serial singularity analysis of delta-like robot is done Using the function **SerialSingularities()** in SIROPA library.

```

1 s1           := SerialSingularities ( robot           :: Manipulator )
2 s1_cart      := Projection          ( s1 , robot:-PoseVariables )
3 s1_art       := Projection          ( s1 , robot:-ArticularVariables )

```

Listing III.2 – Projection of serial singularities in workspace and jointspace

Serial singularities and their projections in workspace and joint space are computed using a Gröbner based elimination method. In Eq III.5, $\det(\mathbf{B})_o$, $\det(\mathbf{B})_h$, $\det(\mathbf{B})_t$ and $\det(\mathbf{B})_u$ are the serial singularities of Orthoglide, Hybridglide, Triaglide and UraneSX, respectively. one can compute (the algebraic closure of) the projection of the parallel singularities in the joint space and workspace. Both are then defined as the zero set of some system of algebraic equations and we assume that the considered robots are generic enough so that both are hypersurfaces. Also these surfaces are shown in Figure III.3 and Figure III.4.

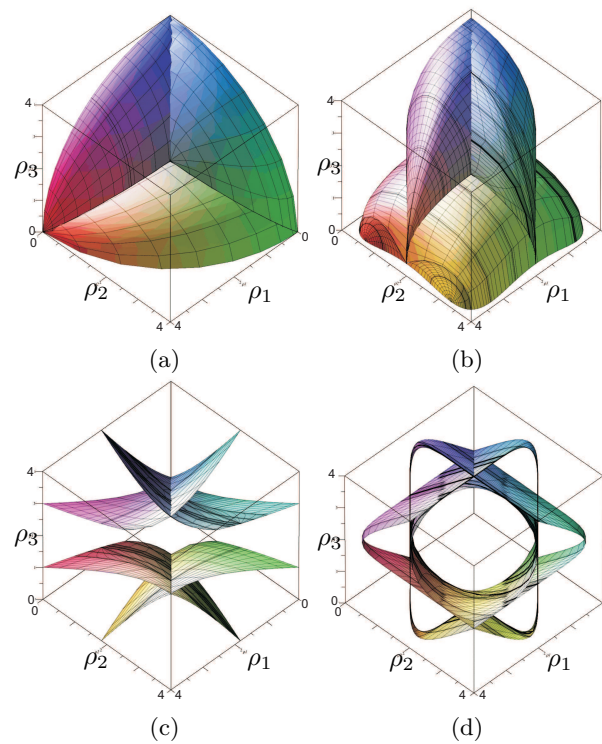


Figure III.3 – Projection of serial singularity curve of Orthoglide (a), Hybridglide (b), Triaglide (c) and UraneSX (d) in jointspace

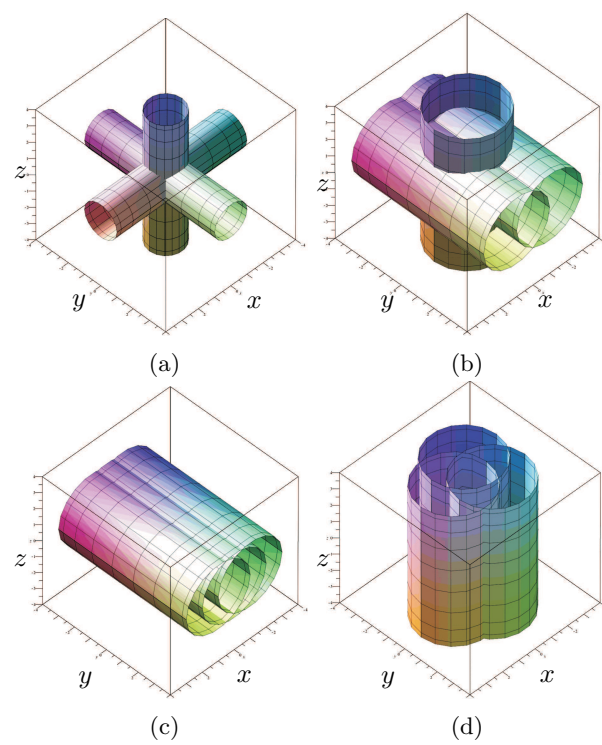


Figure III.4 – Projection of serial singularity curve of Orthoglide (a), Hybridglide (b), Triaglide (c) and UraneSX (d) in workspace

$$\begin{aligned}
\det(\mathbf{B})_o &= 8(\rho_1 - x)(\rho_2 - y)(\rho_3 - z) \\
\det(\mathbf{B})_h &= 8(\rho_1 - y)(\rho_2 - y)(\rho_3 - z) \\
\det(\mathbf{B})_u &= 8(\rho_1 - z)(\rho_2 - z)(\rho_3 - z) \\
\det(\mathbf{B})_t &= 8(\rho_1 - y)(\rho_2 - y)(\rho_3 - y)
\end{aligned} \tag{III.5}$$

Figure III.3 shows the projections of singularity curve $s1$ in jointspace and $s1_art$ is the projection curve. And $s1_cart$ is the projection curve in workspace and is shown in Figure III.4.

III.2.3 Complexity in Singularities

Table III.1 – Comparison of the different parameters associated with the parallel singularities for the robots

PARALLEL SINGULARITIES					
Projection in Workspace					
Manipulators	Plotting Time	Degrees	No.of terms	Binary	No. of Cells
Orthoglide	0037.827	18[10,10,10]	097	015	[02382,0272]
Hybridglide	5698.601	20[16,08,12]	119	017	[28012,1208]
Triaglide	0010.625	03[00,00,03]	002	002	[00138,0004]
UraneSX	0022.625	06[06,04,00]	015	040	[02795,0070]

In Table III.1, a comparative study of five parameters among the family of delta like robot is presented. We have tabulated the main characteristics of the polynomials (In three variables) used for the plots (Implicit surface) : their total degree, their number of terms and the maximum bitsize of their coefficients. We have also reported the time (In seconds) for plotting the implicit surface which they define and the number of cells computed by the CAD, as well as the number of cells in the final result after gluing those that are adjacent and belongs to the same connected component. Several functions are used which involves the discriminant variety, Gröbner bases and CAD computations, computed in Maple 18 with a Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz (14 Gb RAM). As can be seen from Table III.1, there exists higher values of all the parameters for the Hybridglide, among all manipulators listed, which infers that it has more complex parallel singularities, whereas for the Triaglide all the values are least which intuits the less complicated singularities. For example, the computation times for the Hybridglyde for parallel singularities is high compared to the one for the Othoglide, even if the surface has similar characteristics. This is due to the geometry of the surface which is more difficult to decompose in the case of the Hybridglide : the Cylindrical Algebraic Decomposition is described by 1208 cylindrical cells in the case of Hybridglide while it is described by 272 cells for the Orthoglide.

In Table III.2, a comparative study of five parameters among the family of delta like robot is presented. We have tabulated the main characteristics of the polynomials (In three

Table III.2 – Comparison of the different parameters associated with the serial singularities for the robots

SERIAL SINGULARITIES					
Projection in Joint Space					
Manipulators	Plotting Time	Degrees	No.of terms	Binary	No. of Cells
Orthoglide	0005.133	18[12,12,12]	062	012	[00044,0004]
Hybridglide	0007.007	18[12,12,12]	281	017	[00158,0027]
Triaglide	0005.079	06[06,06,06]	042	007	[00077,0017]
UraneSX	0018.391	12[12,12,12]	252	151	[00392,0142]

variables) used for the plots (Implicit surface) : their total degree, their number of terms and the maximum bitsize of their coefficients. We have also reported the time (In seconds) for plotting the implicit surface which they define and the number of cells computed by the CAD, as well as the number of cells in the final result after gluing those that are adjacent and belongs to the same connected component. Several functions are used which involves the discriminant variety, Gröbner bases and CAD computations, computed in Maple 18 with a Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz (14 Gb RAM). As can be seen from Table III.2, there exists higher values of all the parameters for the UraneSX, among all manipulators listed, which infers that it has more complex serial singularities, whereas for the Triaglide all the values are least which intuites the less complicated singularities. The computation times for plotting the serial singularities in jointspace is high compared to others.

III.3 Parallel Singularities of 3RPS Parallel robot

Differentiating the constraints equations with respect to time leads to the velocity model:

$$\mathbf{A}\mathbf{t} + \mathbf{B}\dot{\mathbf{q}} = 0 \quad (\text{III.6})$$

where \mathbf{A} and \mathbf{B} are the parallel and serial Jacobian matrices, respectively, \mathbf{t} is the velocity of P and $\dot{\mathbf{q}}$ joint velocities. The parallel singularities occur whenever $\det(\mathbf{A}) = 0$. Let OM^1 be the operation mode where $q_1 = 0$ and OM^2 , where $q_4 = 0$, then \mathcal{S}_{OM^1} and \mathcal{S}_{OM^2} represent the singularity locus and are characterized by:

$$\begin{aligned} \mathcal{S}_{OM^1} : & q_4(8q_2q_3^2q_4^6 + 2q_2q_4^8 - 64zq_3^6q_4 - 96zq_3^4q_4^3 - 36zq_3^2q_4^5 - 6zq_4^7 \\ & - 24z^2q_2q_3^2q_4^2 - 6z^2q_2q_4^4 - 32q_2q_3^2q_4^4 - 10q_2q_4^6 + 2z^3q_4^3 + 96zq_3^4q_4 \\ & + 72zq_3^2q_4^3 + 23zq_4^5 + 16z^2q_2q_3^2 + 10z^2q_2q_4^2 + 8q_2q_4^4 - z^3q_4 - 36zq_3^2q_4 \\ & - 21zq_4^3 - 4z^2q_2 + 4zq_4) = 0 \end{aligned} \quad (\text{III.7})$$

$$\begin{aligned} \mathcal{S}_{OM^2} : & q_1^2(6q_1^7q_3 + 8q_1^5q_3^3 - 2zq_1^6 + 36zq_1^4q_3^2 + 96zq_1^2q_3^4 + 64zq_3^6 \\ & - 18z^2q_1^3q_3 - 24z^2q_1q_3^3 - 18q_1^5q_3 - 16q_1^3q_3^3 + 2z^3q_1^2 + 3zq_1^4 - 72zq_1^2q_3^2 \\ & - 96zq_3^4 + 18z^2q_1q_3 + 12q_1^3q_3 - z^3 + 3zq_1^2 + 36zq_3^2 - 4z) = 0 \end{aligned} \quad (\text{III.8})$$

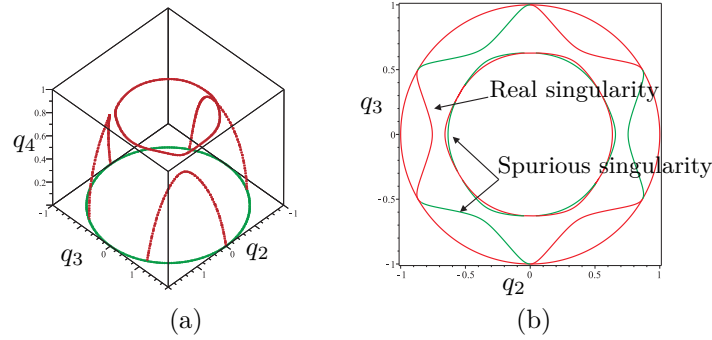


Figure III.5 – Singularity curves for $z = 3$, $q_1 = 0$ (a) and its 2D projection under (q_2, q_3) (b)

The serial singularities occur whenever $\det(\mathbf{B}) = 0$ i.e $\rho_1\rho_2\rho_3 = 0$. The common coordinates for the both operation modes are z , q_2 and q_3 . Figure III.5(a) represents the singularity curve for OM^1 for a fixed value of z (i.e $z = 3$) and Figure III.5(b) shows its projection in the two dimensional coordinate space $(q_2 - q_3)$.

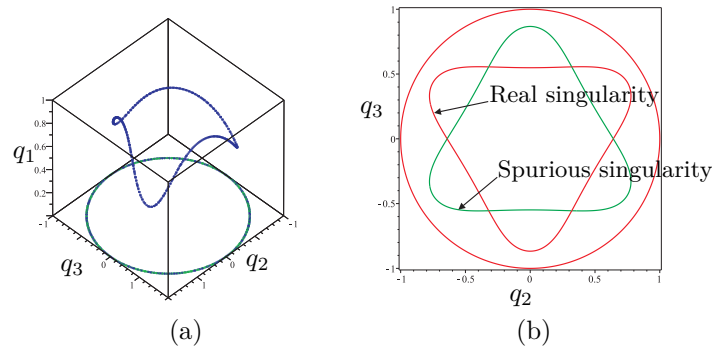


Figure III.6 – Singularity curves for $z = 3$, $q_4 = 0$ (a) and its 2D projection under (q_2, q_3) (b)

Due to the redundancy of the quaternion representation there exists two triplets based on these three coordinates to represent the same pose in the same operation mode. To overcome this problem, we set $q_1 > 0$ and $q_4 > 0$. We can then depict a slice of this surface by fixing one parameter as shown in Figures III.5(a) and III.6(a). Figures III.5(b) and III.6(b) shows the singularity curves in the projection space z , q_2 and q_3 , where the red curve is a real singular locus, which represents the positive values of q_4 or q_1 and the green curve is a spurious singularity curve, which represents the negative values of q_4 or q_1 .

III.4 Influence of design parameter on Parallel Singularities

As there exists parasitic motions for 3RPS parallel robot, it is possible to analyze the singularities in two different projection spaces, $[p_z, q_2, q_3]$ and $[p_x, p_y, p_z]$. Figures III.7 & III.8 represents the projection of parallel singularities in projection space $[p_x, p_y, p_z]$ for OM^1 and OM^2 , respectively. Variation in the parallel singularities surface for different design parameter k is shown in Figures III.7 & III.8. Also the top view of these singularity surfaces is presented to analyze the change in the singularities surface boundaries with the

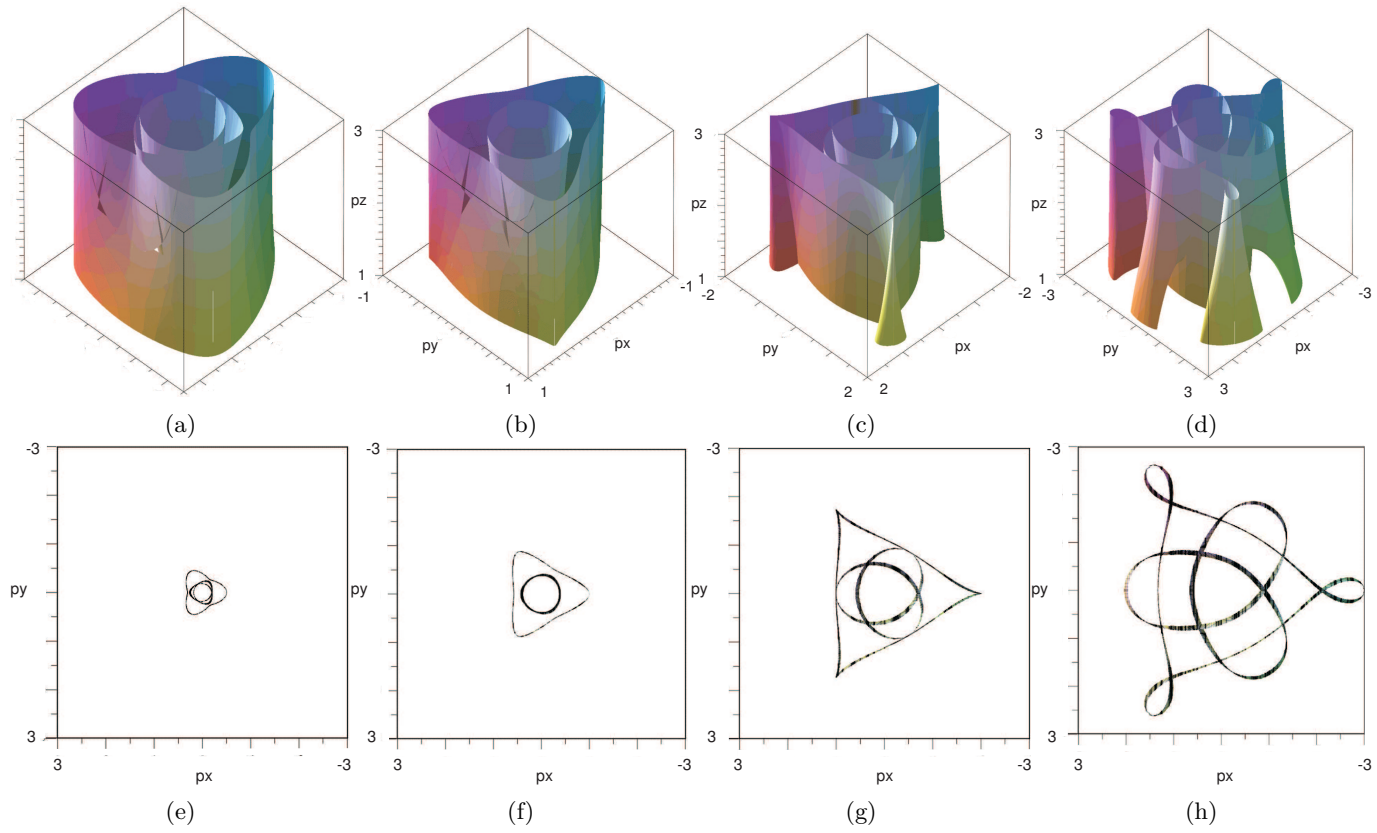


Figure III.7 – Projection of parallel singularities in projection space $[p_x, p_y, p_z]$ for OM^1 . Variation in the parallel singularities surface for different design parameter $k = 0.5$ (a), $k = 1$ (b), $k = 2$ (c) and $k = 3$ (d), top view of the parallel singularities surface, $k = 0.5$ (e), $k = 1$ (f), $k = 2$ (g) and $k = 3$ (h).

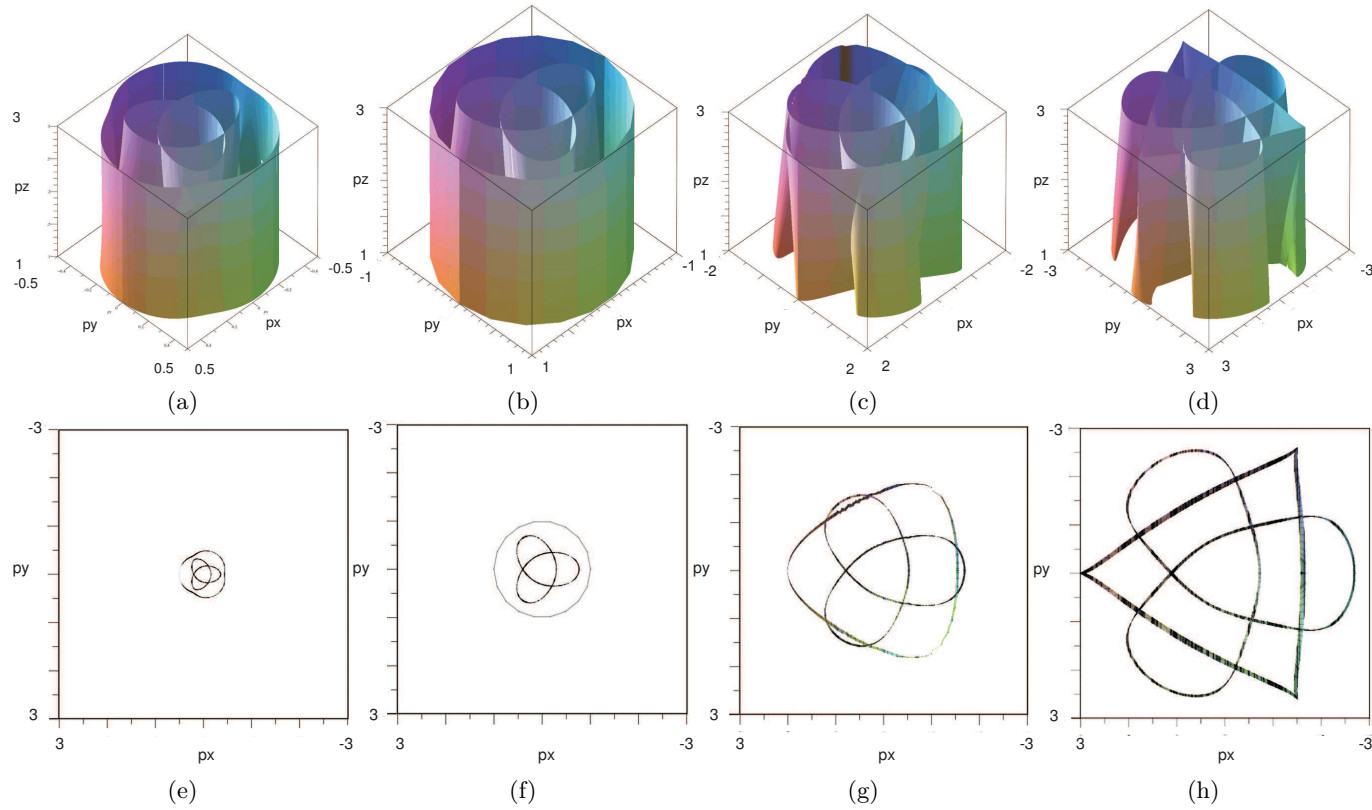


Figure III.8 – Projection of parallel singularities in projection space $[p_x, p_y, p_z]$ for OM^2 . Variation in the parallel singularities surface for different design parameter $k = 0.5$ (a), $k = 1$ (b), $k = 2$ (c) and $k = 3$ (d), top view of the parallel singularities surface, $k = 0.5$ (e), $k = 1$ (f), $k = 2$ (g) and $k = 3$ (h).

change in the value of k . As the value of k increases the connectivity in the workspace region decreases, can be seen in Figures III.7(e-h) & III.8(e-h).

III.5 Aspect for an Operation mode

In [17], the notion of aspect is defined for parallel robots with only one inverse kinematic solution. This notion is now extended for a parallel robot with several operation modes such that:

- $WA_i^j \subset W_{OM^j}$
- WA_i^j is connected
- $\forall X \in WA_i^j, \det(\mathbf{A}) \neq 0$ and $\det(\mathbf{B}) \neq 0$

In other words, an aspect WA_i^j is the maximum connected region without any singularity of the OM^j . The analysis of the workspace is done in the projection space (z, q_2, q_3) , and shows the existence of four aspects as shown in Figure III.9. However, no further analysis

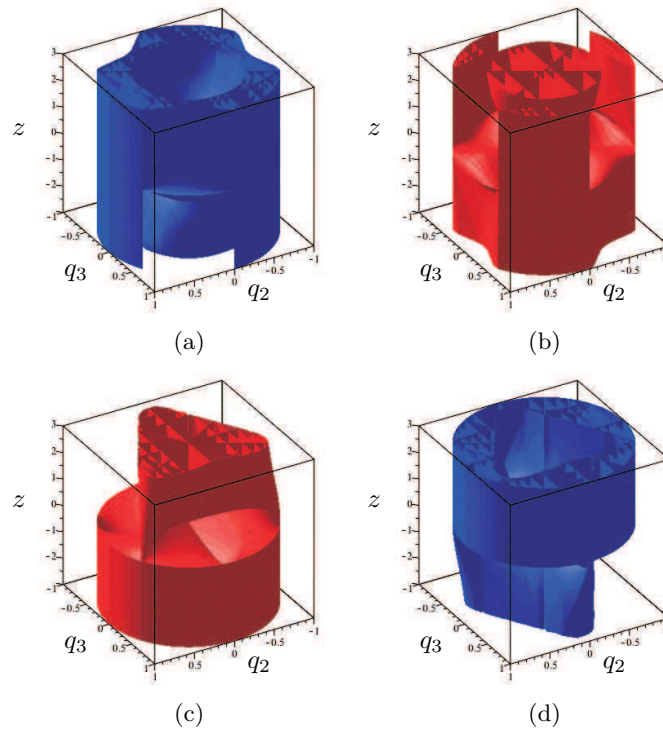


Figure III.9 – Aspects for OM^1 with $\det(\mathbf{A}) < 0$ (a) and $\det(\mathbf{A}) > 0$ (b) and aspects for OM^2 with $\det(\mathbf{A}) < 0$ (c) and $\det(\mathbf{A}) > 0$ (d)

is done to prove this feature in the four dimension space. As there are several solutions for the DKP in the same aspect, non-singular assembly mode trajectories are possible. The cylindrical algebraic decomposition (CAD) implemented in the SIROPA library has been used to decompose an aspect into a set of cells where algebraic equations define its boundaries [12]. The CAD provides a formal decomposition of the parameter space in cells where the polynomials $\det(\mathbf{A})$ and $\det(\mathbf{B})$ have a constant sign [73] and the number of solutions for the DKP is constant.

III.6 Characteristic surfaces for an operation mode

The notion of *characteristic surface* was introduced in [84] to define the uniqueness domains for serial Cuspidal robots. This definition was extended to parallel robots with one inverse kinematic solution in [17] and with several inverse kinematic solutions in [77, 12]. In this paper, we introduce this notion for a parallel robot with several operation modes and a single working mode.

Let WA_i^j be one aspect for the operation mode j . The characteristic surfaces, denoted by $\mathcal{S}_C^j(WA_i^j)$, are defined as the preimage in WA_i^j of the boundary $\overline{WA_i^j}$ that delimits WA_i^j .

$$\mathcal{S}_C^j(WA_i^j) = g_j^{-1} \left(g_j(\overline{WA_i^j}) \right) \cap WA_i^j \quad (\text{III.9})$$

These characteristic surfaces are the images in the workspace of the singularity surfaces. By using the singularity and characteristic surfaces, we can compute the *basic regions* as defined in [17].

Starting from the Figures III.5(a) and III.6(a), Figure III.10 depicts a slice of the singularity curves and characteristic surfaces for $z = 3$. We cannot factorize the polynomial equation of the characteristic surface to separate the positive and negative values of q_4 for the OM^1 and q_1 for the OM^2 .

The basic regions WAb_i^j of the operation mode j are calculated as the connected components of the set obtained by removing the characteristic surfaces from the aspects WA_i^j :

$$\cup WAb_i^j = WA_i^j - \mathcal{S}_C^j \quad (\text{III.10})$$

The singularity surfaces split the joint space in regions where the number of solutions to

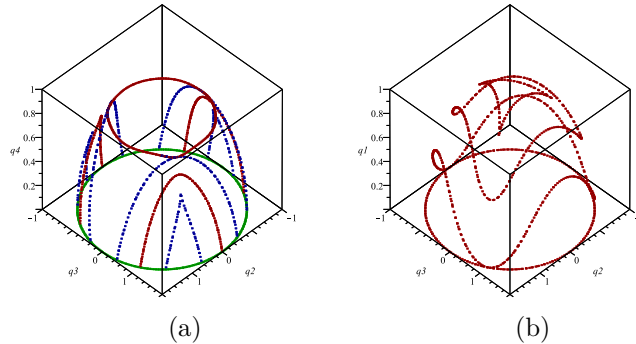


Figure III.10 – Singularity curves and characteristic surfaces with $z = 3$ for OM^1 (a) OM^2 (b)

the DKP is constant. We name these regions as the *basic components* as in [17].

The basic components are computed of the operation mode j as:

$$QA_i^j = g_j \left(WAb_i^j \right) \quad (\text{III.11})$$

III.7 Uniqueness domains

we only report a slice of the workspace to define the uniqueness domains. Nevertheless, the reader has to keep in mind that some regions are connected in the 3D space. For $z = 3$,

the basic regions are computed by using the cylindrical algebraic decomposition for both aspects and operation modes. For the computation of aspects, the CAD provides a set of cells with a test point. To find the property of these aspect we have to solve the DKP followed by IKP.

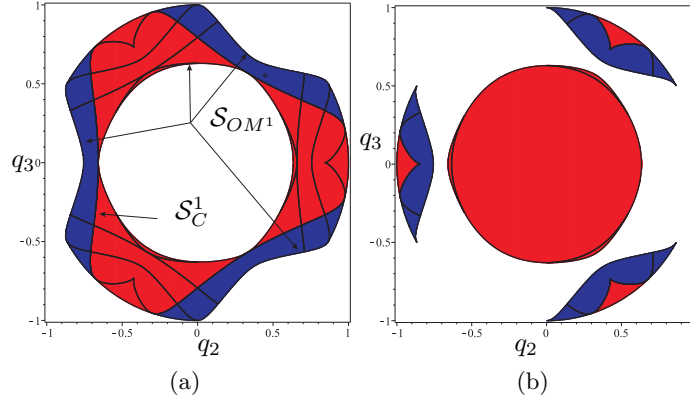


Figure III.11 – Slice of an aspect for $z = 3$ and OM^1 for $\det(\mathbf{A}) > 0$ (a) $\det(\mathbf{A}) < 0$ (b) with basic regions in blue (reps. in red) coming from a basic component with four DKP (resp. eight)

In Figure III.17, the slice of an aspect at $z = 3$ and for both positive and negative values of $\det(\mathbf{A})$ is shown for the operation mode OM^1 . Also the slice of an aspect for the operation mode, OM^2 is shown in Figure III.12, where the blue and red basic regions represents the basic component with four and eight to the DKP. It can be observed that the red regions in Fig. III.17a (respectively, III.12a) have a contact point which is a tangency between \mathcal{S}_{OM^1} and \mathcal{S}_C^1 (respectively, \mathcal{S}_{OM^2} and \mathcal{S}_C^2). These points are the image of the cusp points in the workspace, i.e. a triple root for the DKP.

In the slice at $z = 3$, there is 13 and 5 basic regions for OM^1 and OM^2 . For OM^1 , the basic regions coming from a basic component where we have 8 solutions for the DKP are disconnected from each other. However, this feature exists only in the slice and the basic regions are connected on the 3D space. We can note that there are curves which intersect the base regions. These curves are the components above negative values of q_1 and q_4 , singularities and characteristic surfaces. One then must make use of the CAD to figure out which curves are relevant in the projective space.

The uniqueness domains Wu_k^j for the operation mode j are the union of two sets: the set of adjacent basic regions $\cup_{i \in I'} WA_i^j$ of the same aspect WA_i^j whose respective preimages are disjoint basic components, and the set $\mathcal{S}_C^j(I')$ of the characteristic surfaces which separate these basic components:

$$Wu_k^j = \left(\cup_{i \in I'} WA_i^j \right) \cup \mathcal{S}_C^j(I') \quad (\text{III.12})$$

with $I' \subset I$ such as $\forall i_1, i_2 \in I', g_j(WAb_{i_1}^j) \cap g_j(WAb_{i_2}^j) = \emptyset$.

The proof for the uniqueness definition is given in [17].

Figure III.13(a) represents the basic regions of OM^1 for $\det(\mathbf{A}) > 0$ with all the uniqueness domains shown in Figure III.13(b)-(d). The associated uniqueness domains can be

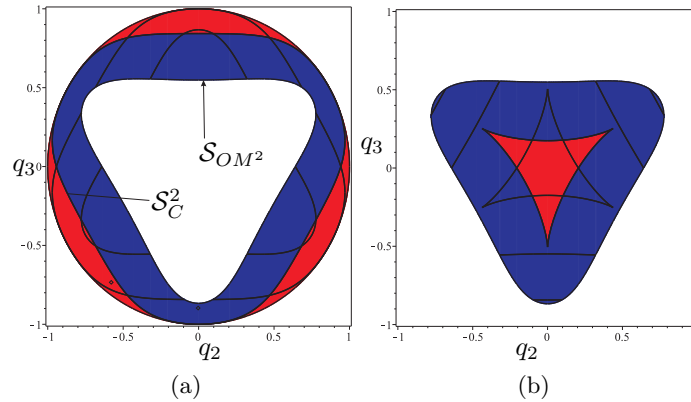


Figure III.12 – Slice of an aspect for $z = 3$ and OM^2 for $\det(\mathbf{A}) > 0$ (a) $\det(\mathbf{A}) < 0$ (b) with basic regions in blue (reps. in red) coming from a basic component with four DKP (resp. eight)

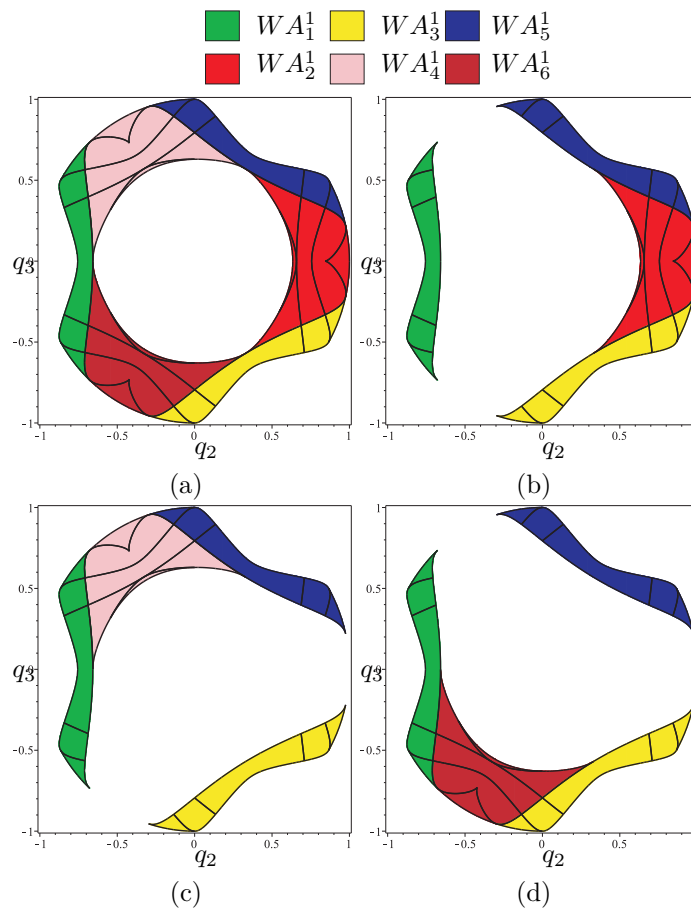


Figure III.13 – Basic regions of OM^1 for $\det(\mathbf{A}) > 0$ (a), associated uniqueness domains $Wu_1^1 = WA_1^1 \cup WA_2^1 \cup WA_3^1 \cup WA_4^1$ (b), $Wu_2^1 = WA_1^1 \cup WA_2^1 \cup WA_3^1 \cup WA_5^1$ (c) and $Wu_3^1 = WA_1^1 \cup WA_2^1 \cup WA_3^1 \cup WA_6^1$ (d)

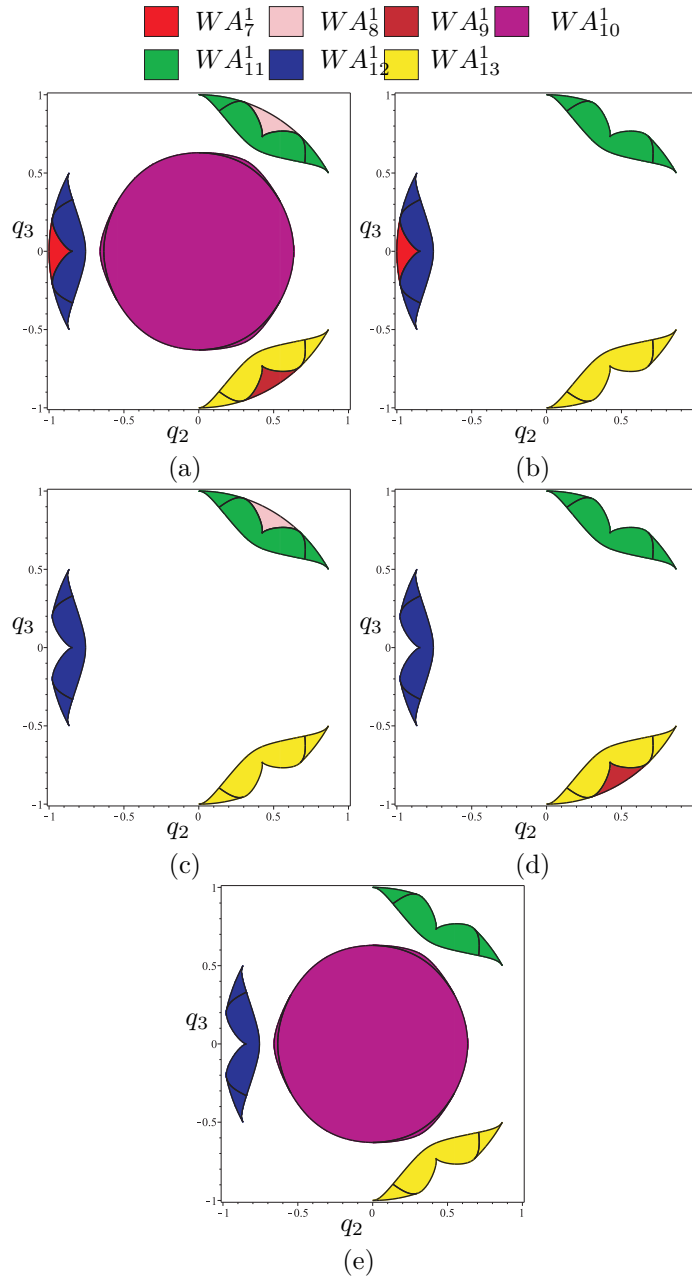


Figure III.14 – Basic regions of OM^1 for $\det(\mathbf{A}) < 0$ (a), associated uniqueness domains $Wu_4^1 = WA_7^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1$ (b), $Wu_5^1 = WA_8^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1$ (c), $Wu_6^1 = WA_9^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1$ (d) and $Wu_7^1 = WA_{10}^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1$ (e)

represented as:

$$\begin{aligned}
 Wu_1^1 &= WA_1^1 \cup WA_2^1 \cup WA_3^1 \cup WA_4^1 & (III.13) \\
 Wu_2^1 &= WA_1^1 \cup WA_2^1 \cup WA_3^1 \cup WA_5^1 \\
 Wu_3^1 &= WA_1^1 \cup WA_2^1 \cup WA_3^1 \cup WA_6^1
 \end{aligned}$$

The various aspect of the operation modes are represented by WA_j^i where i denotes the operation mode and j represents the basic regions. The same illustration is shown in Fig-

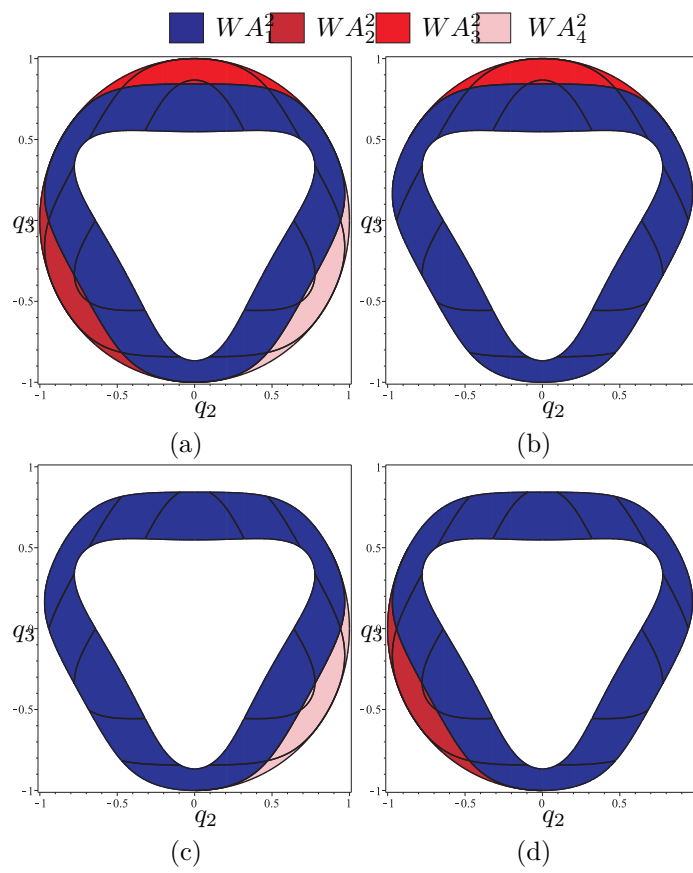


Figure III.15 – Basic regions of OM^2 for $\det(\mathbf{A}) > 0$ (a), associated uniqueness domains $Wu_1^2 = WA_1^2 \cup WA_2^2$ (b), $Wu_2^2 = WA_1^2 \cup WA_3^2$ (c) and $Wu_3^2 = WA_1^2 \cup WA_4^2$ (d)

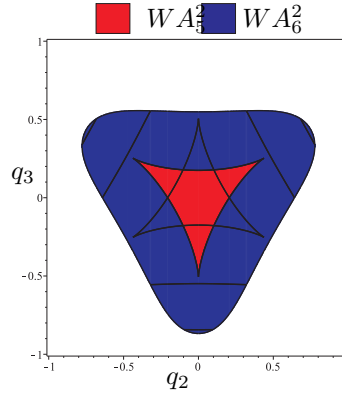


Figure III.16 – Basic region of OM^2 for $\det(\mathbf{A}) < 0$ or the uniqueness domain $Wu_4^2 = WA_5^2 \cup WA_6^2$

ure III.14 for OM^1 with $\det(\mathbf{A}) < 0$.

$$\begin{aligned}
 Wu_4^1 &= WA_7^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1 & (III.14) \\
 Wu_5^1 &= WA_8^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1 \\
 Wu_6^1 &= WA_9^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1 \\
 Wu_7^1 &= WA_{10}^1 \cup WA_{11}^1 \cup WA_{12}^1 \cup WA_{12}^1
 \end{aligned}$$

Similarly, Figures III.15 and III.16 represent the basic regions of OM^2 and the uniqueness domains associated for $\det(\mathbf{A}) > 0$ and $\det(\mathbf{A}) < 0$, respectively.

$$\begin{aligned}
 Wu_1^2 &= WA_1^2 \cup WA_2^2 & (III.15) \\
 Wu_2^2 &= WA_1^2 \cup WA_3^2 \\
 Wu_3^2 &= WA_1^2 \cup WA_4^2 \\
 Wu_4^2 &= WA_5^2 \cup WA_6^2
 \end{aligned}$$

III.8 Non Singular assembly mode Trajectories

Letting $z = 3$, the basic regions are computed by using the cylindrical algebraic decomposition for a given aspect. Figure III.17 shows the three basic regions' image of basic components with 8 solutions for the DKP and a single basic region's image of a basic components with 4 solutions for the DKP connects these three previous basic regions. Table III.3 presents the roots of the DKP for $\det(\mathbf{A}) > \mathbf{0}$ for a joint position in each operation mode. For each of them, we find out that three roots have their z coordinate close to 3. A non-singular assembly mode changing trajectory can be obtained between three basic regions coming from eight solutions to the DKP. Due to symmetrical properties, there are also three roots of the DKP for $\det(\mathbf{A}) < \mathbf{0}$ with $z = -3$. For OM^1 , we construct a path between P_1, P_2, P_3 and for OM^2 between P_5, P_6 and P_7 . When a straight line between two poses cross a singularity, we add an intermediate point as shown in Fig. III.17. The connections between the basic regions depicted in red are the projections of the cusp points in the workspace, i.e. the tangent between the singularity surface and the characteristic surface [12].

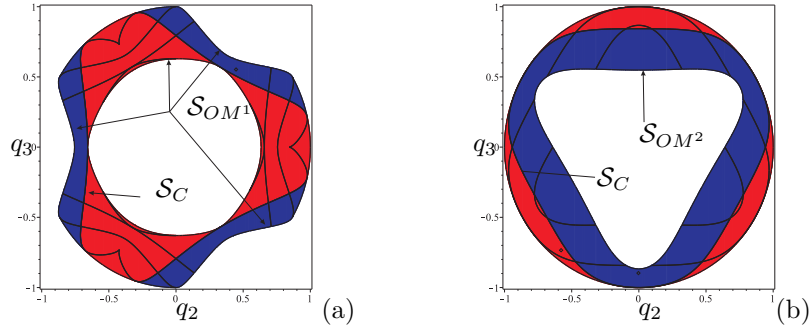


Figure III.17 – Slice of an aspect for $z = 3$ and $\det(\mathbf{A}) > 0$ for OM^1 (a) and OM^2 (b) with in blue (resp. in red) a basic region coming from a basic component with four DKP (resp. eight)

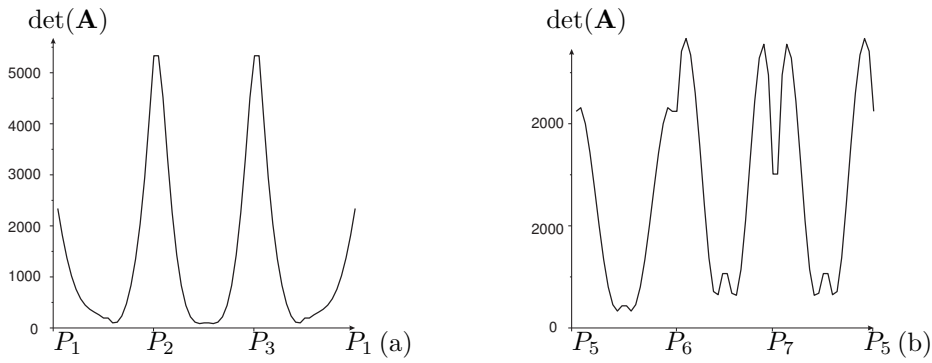


Figure III.18 – Variation of $\det(\mathbf{A})$ along trajectory P_1, P_2, P_3 for OM^1 (a) and P_5, P_6, P_7 for OM^2 (b)

Table III.3 – Solutions of the DKP for $\det(\mathbf{A}) > 0$

OM^1					OM^2				
$\rho_1 = 3.90, \rho_2 = 3.24, \rho_3 = 3.24$					$\rho_1 = 3.79, \rho_2 = 3.24, \rho_3 = 3.24$				
P	z	q_2	q_3	q_4	P	z	q_1	q_2	q_3
P_1	3.01	-0.34	-0.94	0.06	P_5	3.04	0.35	-0.58	-0.74
P_2	3.01	-0.34	0.94	0.06	P_6	3.04	0.35	0.586	-0.74
P_3	3	0.85	0.0	0.53	P_7	3	0.24	0.0	0.97
P_4	-2.88	-0.35	0.0	0.93	P_8	-3.42	0.98	0.0	0.19

The variation of the $\det(\mathbf{A})$ is plotted in the Figure III.18 and shows the existence of a non-singular assembly mode changing trajectory. The image of this trajectory in the joint space is illustrated in the Figure III.19. The projection of the cyclic trajectory defined by (P_1, P_2, P_3, P_1) (resp. (P_5, P_6, P_7, P_5)) onto the joint space encloses three curves of cusps.

This behavior is similar to that of the 3-RPR robot described in [70] or the 3-RPS robot in [79]. The path to connect the fourth solutions is not presented in this paper. The method introduced in [13] is used to compute the cusp curves.

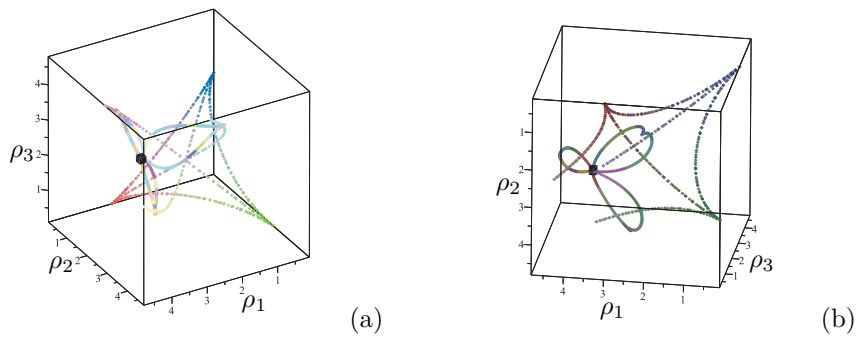


Figure III.19 – Projection in Q of the trajectories with the cusp curves for OM^1 (a) and OM^2 (b)

III.9 Conclusions

In this chapter, we have studied the singularities of the 3-RPS parallel robot and delta like family robot. The main characteristics associated with the singularities of the delta like family robot are tabulated in Table III.1 & III.2, which also gives some information about the complexity of the singularities, which is an essential factor for the singularity-free path plannings. From these data, it can be observed that the singularities associated with the Hybridglide are complicated, whereas the structure of those associated with the Triaglide is rather simple.

With the presented two operation modes, the aspects were set to define the maximal singularity free region of the workspace. Each operation mode is divided into two aspects which further show that non-singular assembly mode changing trajectories are possible. Basic regions can be defined for each operation mode by computing the characteristic surfaces. With the knowledge of basic regions defined in the workspace, the uniqueness domains can be illustrated in a slice of the workspace. This method can be used for the optimization of workspace's size or the trajectory planning.

IV

Trajectory Planning

Trajectory planning is a critical step while programming the parallel robots in a robotic cell. The main problem arises when there exists a singular configuration between two poses of the end-effectors while discretizing the path with a classical approach. This Chapter presents the joint space analysis and an algebraic method to check the feasibility of any given trajectories in the workspace. The solutions of the polynomial equations associated with the trajectories are projected in the joint space using Gröbner based elimination methods and the remaining equations are expressed in a parametric form where the articular variables are functions of time t unlike any numerical or discretization method. These formal computations allow to write the Jacobian of the manipulator as a function of time and to check if its determinant can vanish between two poses. Another benefit of this approach is to use a largest workspace with a more complex shape than a cube, cylinder or sphere.

IV.1 Introduction

One of the crucial steps in the trajectory planning is to check the singularity-free paths in the workspace for the parallel robots. It becomes a necessary protocol to validate the trajectory when the parallel robot is used in practical applications such as precise manufacturing operations. A trajectory verification problem is presented in [25] based on some validity criteria like whether the trajectory lies fully inside the workspace of the robot and is singularity-free. Singularity-free path planning for the Gough-Stewart platform with a given starting pose and a given ending pose has been addressed in [26] using the clustering algorithm is presented in [27]. An exact method and an approximate method are described in [28] to restructure a path close to the singularity locus into a path that avoids it while remaining close to the original path. Due to the geometry of the mechanism, the workspace may not cover fully the space of poses [27], hence it is necessary to analyze the workspace of the manipulator.

The workspace of a parallel robot mainly depends on the actuated joint variables, the range of motion of the joints and the mechanical interferences between the bodies of the mechanism. There are different techniques based on geometric tools [3, 4], discretization [5, 7, 8], and algebraic methods [9, 10, 11, 12] which are used to compute the workspace of

a parallel robot. An algebraic method to solve the forward kinematics problem specifically applied to spatial parallel manipulators is described in [29]. The main advantage of the geometric approach is that, it establishes the nature of the boundary of the workspace [19]. A procedure to automatically generate the kinematic model of parallel mechanisms which further used for singularity free path planning is reported in [30].

An algorithm for computing singularity-free paths on closed-chain manipulators is presented in [31], also this method attempts to connect the given two non-singular configurations through a path that maintains a minimum clearance with respect to the singularity locus at all points. The main drawback of any numerical or discretization methods is that there might be a singular configuration between two poses of the end-effectors while discretizing the path. This chapter of the thesis illustrates a technique based on some algebraic methods to check the feasibility of any given trajectories in the workspace : it allows to write the Jacobian of the manipulator as a function of time and to check whether its determinant vanishes between two poses. Also, when the trajectory meets a singularity, its location can also be computed.

IV.2 Methodology

To ensure the non existence of a singular configurations between two poses of the end-effector, it is necessary to express the Jacobian of the manipulator as a function of the time. The general procedure to check the feasibility of the trajectories in the workspace as well as a function to compute the projection of the trajectories in the joint space can be decomposed as follows :

- Defining the constraint equations, articular and pose variables associated with the parallel mechanism;
- Computing the singularities and their projections in the workspace and in the joint space;
- Computing the workspace and joint space boundaries;
- Computing a parametric form of the trajectories in the Cartesian space;
- Computing the projections of the trajectories in the joint space;
- Computing the Jacobian of the manipulator as a function of the time.

By computing, we mean, by default, getting a full characterization as solutions of an exact system of algebraic equations. By abuse of language, computing a projection might thus mean computing the algebraic closure of the projection, for example by using classical elimination strategies based on Gröbner bases (which thus could introduce a subset of null measure made of spurious points).

Kinematics involves the study of the position, velocity, acceleration, and all higher order derivatives of the position variables (with respect to time or any other parameter/variables). Hence, the study of the kinematics of manipulators refers to all the geometrical and

time-based properties of the motion. For a translational manipulator, the set of (algebraic) relations that connects the input values $\boldsymbol{\rho}$ to the output values \mathbf{X} will be denoted by

$$F(\boldsymbol{\rho}, \mathbf{X}) = 0 \quad (\text{IV.1})$$

where $\boldsymbol{\rho}$ is the set of all actuated joint variables and \mathbf{X} is the set of all pose variables of the end-effector.

For the study of manipulators, two problems must be considered: the direct kinematic problem (DKP) and the inverse kinematic problem (IKP). Specifically, given a set of joint values, solving the DKP consists in computing the position and the orientation of the end-effector relatively to the base, or, equivalently to change the representation of the manipulator position from a joint space description into a Cartesian space description. Given the position and orientation of the end-effector of the manipulator, solving the IKP problem consists in computing all the possible sets of joint angles or parameters that could be used to attain this position and orientation, or, equivalently, mapping locations in three-dimensional Cartesian space to locations in the robot's internal joint space.

When considering an algebraic modelization, the direct kinematics has basically several solutions, which refers to the several poses of the end-effector for given values of the joint coordinates. It is therefore possible to assemble the manipulator in different ways, and these different configurations are known as the assembly modes of the manipulator [47]. Similarly, the multiple inverse kinematic solutions induce multiple postures for each leg of the manipulator and is termed as the working modes of the manipulator. An analytical relation can be given as $\rho = \gamma(\mathbf{X})$ for IKP whereas $\mathbf{X} = \beta(\rho)$ for DKP. Differentiating Eq. (IV.1) with respect to time leads to the velocity model:

$$\mathbf{A}\dot{\mathbf{X}} + \mathbf{B}\dot{\boldsymbol{\rho}} = \mathbf{0} \quad \text{where} \quad \mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{X}}, \quad \mathbf{B} = \frac{\partial \mathbf{F}}{\partial \boldsymbol{\rho}} \quad (\text{IV.2})$$

The matrices \mathbf{A} and \mathbf{B} are respectively the direct-kinematics and the inverse-kinematics Jacobian matrices of the manipulator. These matrices are used for characterizing different kinds of singularities. The parallel singularities occur whenever $\det(\mathbf{A}) = 0$ and the serial singularities occur whenever $\det(\mathbf{B}) = 0$. The parallel singular configurations are located inside the workspace. They are particularly undesirable because the manipulator cannot resist to any forces and its control is lost.

Eliminating ρ in the system $F(\rho, X) = 0$, $\det(\mathbf{A}) = 0$ by means of a Gröbner basis computation for a suitable elimination ordering (see [83]) defines (the algebraic closure of) the projection $\xi(\mathbf{X})$ of the parallel singularities in the workspace. In the same way, one can compute (the algebraic closure of) the projection $\varepsilon(\rho)$ of the parallel singularities in the joint space. Both are then defined as the zero set of some system of algebraic equations and we assume that the considered robots are generic enough so that both are hypersurfaces.

The set of equations associated with the joint limits of the actuator $\chi(\rho)$ can also be projected, with the same elimination method, in the workspace $\mu(\mathbf{X})$ as in Eq. (IV.3), where ρ_{min} , ρ_{max} are the minimum and maximum values of the articular variables: $\chi(\rho)$ and $\mu(\mathbf{X})$ are the crucial parameters in determining the number of assembly modes and working modes of the parallel manipulator.

$$\chi(\rho) \mapsto \mu(\mathbf{X}) \quad \forall \rho \in (\rho_{\min}, \rho_{\max}] \quad (\text{IV.3})$$

To use algebraic tools such as Gröbner bases, we must represent the trajectories in an algebraic form. A classical approach can be used to transform the trigonometric equations to algebraic ones as in [75]. Equation (V.8) represents the trajectory in parametric form within the workspace as the function of time t .

$$\mathbf{X} \mapsto \phi(\mathbf{t}) \quad (\text{IV.4})$$

$\Psi(\mathbf{X}, \rho, \mathbf{t})$ in Eq. (IV.5) is the system of equations which contains kinematic equations $F(\rho, \mathbf{X})$ and the parametric equations of trajectory $\phi(\mathbf{t})$.

$$\Psi(\mathbf{X}, \rho, \mathbf{t}) = [\phi(\mathbf{t}) - \mathbf{X}, \mathbf{F}(\rho, \mathbf{X})] \quad (\text{IV.5})$$

This system of equations is projected in the joint space as $\Upsilon(\rho, \mathbf{t})$ as shown in Eq. (IV.6) using Gröbner based elimination method.

$$\Psi(\mathbf{X}, \rho, \mathbf{t}) \mapsto \Upsilon(\rho, \mathbf{t}) \quad (\text{IV.6})$$

The parametric equations of the trajectory as a function of the time in the joint space can then be obtained by solving $\rho \leftarrow \Upsilon(\rho, \mathbf{t})$.

The workspace analysis allows the characterization of the workspace regions where the number of real solutions for the inverse kinematics is constant. A cylindrical algebraic decomposition (CAD) algorithm is used to compute the workspace of the robot in the projection space \mathbf{X} with joint constraints $\chi(\rho)$ taken in account [16, 52, 56].

The three main steps involved in the analysis are:

- Computation of a subset of the joint space (resp. workspace) where the number of solutions changes: the *Discriminant Variety*.
- Description of the complementary of the discriminant variety in connected cells: the *Generic Cylindrical Algebraic Decomposition* (CAD).
- Connection of the cells belonging to the same connected component in the counterpart of the discriminant variety: *interval comparisons*.

The joint space analysis predicts the feasible and non-feasible combinations of the prismatic joint variables which are essential for the parallel robot control. The joint space analysis allows the characterization of the regions where the number of real solutions for the direct kinematic problem is constant. The joint space analysis is done using CAD, which gives the number of cells corresponding to the number of solutions in the joint space. $\chi(\rho)$ parameter significantly changes the number of assembly modes and working modes of the manipulator.

Substituting $\phi(\mathbf{t})$ in $\xi(\mathbf{X})$ (in the polynomial defining $\xi(\mathbf{X})$) we then get $\xi(\phi(\mathbf{t}))$, which defines the vanishing values of the Jacobian as a function of the time. This equation in $\cos(t), \sin(t)$ can be turned into a zero-dimensional bivariate system and its solutions can

thus be computed exactly, either by means of a rational parametrization or by means of isolating intervals with rational bounds that can be refined to any arbitrary precision ¹.

Whatever the chosen (exact) representation for a solution, it can easily be checked if it vanishes between the two poses of the end-effector. The solutions of $\xi(\phi(\mathbf{t}))$ contain the singular points on the trajectory $\phi(\mathbf{t})$ and eventually few spurious points due to the projection of $\det(\mathbf{A})$ in Cartesian space. Spurious singular points can then be differentiated from real singular points by substituting $\phi(\mathbf{t})$ and the solutions of $\Upsilon(\rho, \mathbf{t})$ in $\det(\mathbf{A})$.

IV.3 Method Validation

We propose to validate our approach (and related tools) by checking the feasibility of three different trajectories for the Orthoglide. The first two trajectories are heart shaped planar parametric curves and the third trajectory is a parametric curve in three dimensions.

These trajectories are fully inside the workspace and are selected such that their projections in the joint space is defined by parametric equations containing trigonometric functions. In addition, we have shown that the singular and singular-free trajectories are well discriminated, as one of the trajectory cuts the singularity surface.

IV.3.1 Manipulator Architecture and Kinematics

The manipulator under the study is an **Orthoglide** parallel robot with three degrees of freedom. The mechanism is driven by three actuated orthogonal prismatic joints ρ_i , made of three parallelogram connected by revolute joints to the tool center point on one side and to the corresponding prismatic joint at another side. The assembly modes of these robots depends on the solutions of the DKP as shown in Fig. IV.1(a). The point \mathbf{P}_i represents the pose of corresponding robot. However more than one value of i for the point \mathbf{P}_i denote multiple solutions for the DKP. $\mathbf{A}_i\mathbf{B}_i$ is equal to ρ_i , where ρ_i represents the prismatic joint variables whereas \mathbf{P} represents the position vector of the tool center point.

The constraint equations for the Orthoglide with the actuated variables $\rho = [\rho_1, \rho_2, \rho_3]$ and the pose variables $\mathbf{X} = [x, y, z]$ are:

$$\begin{aligned} F(\rho, \mathbf{X}) : (x - \rho_1)^2 + y^2 + z^2 &= l^2 \\ x^2 + (y - \rho_2)^2 + z^2 &= l^2 \\ x^2 + y^2 + (z - \rho_3)^2 &= l^2 \end{aligned} \tag{IV.7}$$

where l is the leg length of the manipulator and is equal to two for all the computations. The set of equations associated with the joint limits of the actuator $\chi(\rho)$, projected in the workspace $\mu(\mathbf{X})$, are given in Eq. (IV.8). $\chi(\rho)$ plays an important role in determining the shape of the workspace and singularity surfaces. It also affects the number of solutions for the IKP *i.e.* the working modes associated with the manipulator. Figure IV.1(b) represents the workspace of the Orthoglide. A CAD algorithm is used to compute the workspace of

¹for example the functions Groebner[RationalUnivariateRepresentation] and RootFinding[Isolate] in *Maple* software

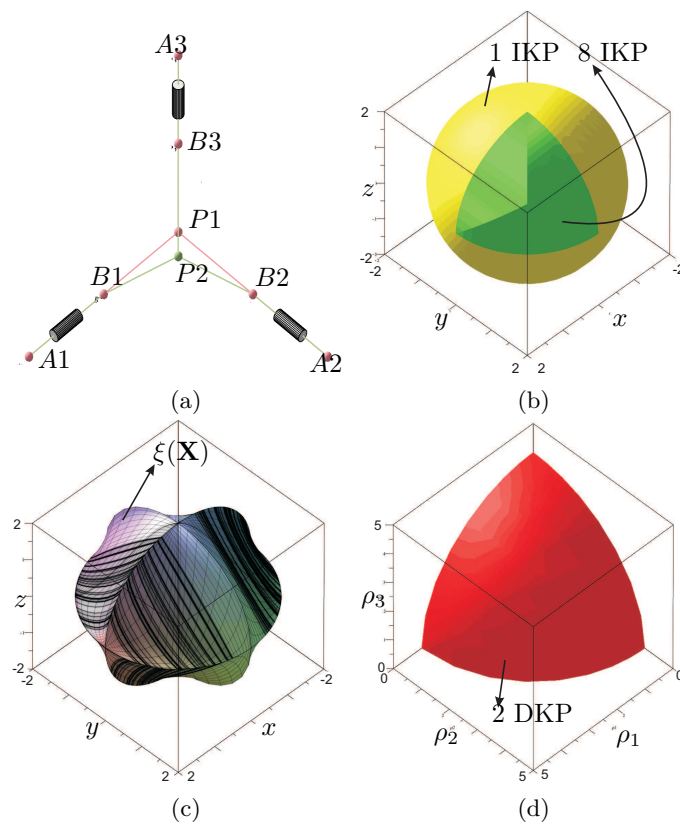


Figure IV.1 – Architecture of an Orthoglide including two assembly modes (a) Workspace plot with $\chi(\rho)$ joint constraints (b) and Parallel singularity $\det(\mathbf{A})$ projected in the workspace as $\xi(\mathbf{X})$ (c) and joint space plot with $\chi(\rho)$ joint limits (d)

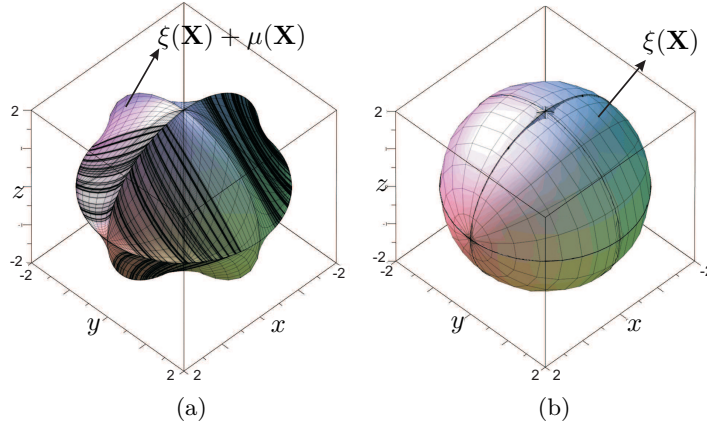


Figure IV.2 – A comparison between the parallel singularity surface for an Orthoglide computed with the joint limits $\mu(\mathbf{X})$ (a) without $\mu(\mathbf{X})$ (b)

the robot in the projection space (x, y, z) , taking into account the joint constraints $\chi(\rho)$. Without considering the joint limits, the Orthoglide admits two assembly modes and eight working modes [76].

$$\begin{aligned}
 \chi(\rho) &= [\rho_1, \rho_2, \rho_3, -4 + \rho_1, -4 + \rho_2, -4 + \rho_3] \\
 \chi(\rho) &\mapsto \mu(\mathbf{X}) \\
 \rho_i &\mapsto x^2 + y^2 + z^2 - 4 \quad i = 1, 2, 3 \\
 -4 + \rho_i &\mapsto x^2 + y^2 + z^2 - 8x + 12 \\
 -4 + \rho_i &\mapsto x^2 + y^2 + z^2 - 8y + 12 \\
 -4 + \rho_i &\mapsto x^2 + y^2 + z^2 - 8z + 12
 \end{aligned} \tag{IV.8}$$

In the Figure IV.2(b), the yellow region (resp. green region), corresponds to the region where the inverse kinematic model has two real solutions (resp. height).

The joint space analysis is done using CAD to find the regions where the direct kinematics problem admits a fixed number of real solutions. For example, in Fig. IV.1(d), the red cell corresponds to to the region where the DKP has two solutions.

There is a difference in the shape of the singularity surface, depending if we consider the joint constraints $\mu(\mathbf{X})$ or not (Fig. IV.2).

$$\begin{aligned}
 \det(\mathbf{A}) &= -8\rho_1\rho_2\rho_3 + 8\rho_1\rho_2z + 8\rho_1\rho_3y + 8\rho_2\rho_3x \\
 \det(\mathbf{A}) &\mapsto \varepsilon(\rho) \quad \det(\mathbf{A}) \mapsto \xi(\mathbf{X}) \\
 \varepsilon(\rho) &= \rho_1^4\rho_2^2 + \rho_1^4\rho_3^2 + \rho_1^2\rho_2^4 + 2\rho_1^2\rho_2^2\rho_3^2 + \rho_1^2\rho_3^4 + \\
 &\quad \rho_2^4\rho_3^2 + \rho_2^2\rho_3^4 - 16\rho_1^2\rho_2^2 - 16\rho_1^2\rho_3^2 - 16\rho_2^2\rho_3^2
 \end{aligned} \tag{IV.9}$$

In Eq. (V.5), $\det(\mathbf{A})$ is the parallel singularity of the Orthoglide and $\xi(\mathbf{X})$ is the projection of $\det(\mathbf{A})$ in workspace. The mathematical expression for $\xi(\mathbf{X})$ is not displayed in Eq. (V.5) due to the lack of space. Fig. IV.1(c) shows the projection $\xi(\mathbf{X})$ which is plotted with

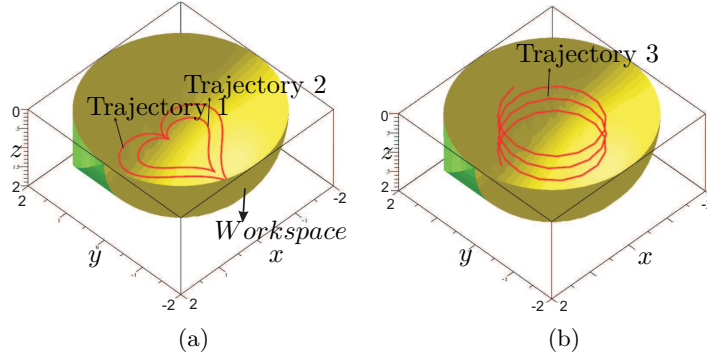


Figure IV.3 – Position of Trajectories 1&2 (a) Trajectory 3 (b) in the workspace of an Orthoglide

$\mu(\mathbf{X})$ as one of the input parameter. The degree of this characteristic surface is 18 and it represents the singularities associated with the eight working modes.

IV.3.2 Trajectory definition in the workspace

Trajectories 1 & 2 are heart shaped parametric curves which (Fig. IV.3(a)). Eq. (IV.10) and Eq. (IV.11) are the mathematical definitions of Trajectory 1 and Trajectory 2, respectively. As these equations are trigonometric equations, it is necessary to represent them in an algebraic form. $\phi_1(t)$ and $\phi_2(t)$ are the parametric equations for Trajectory 1 and Trajectory 2, respectively, which are defined for $t \in [-\pi, \pi]$. From now, sin and cos are replaced by **s** and **c** respectively to reduce the size of the expressions.

$$\begin{aligned} \phi_1(t) : \quad & x = \frac{8}{7}\mathbf{s}^3(t) \\ & y = \frac{13}{14}\mathbf{c}(t) - \frac{5}{14}\mathbf{c}(2t) - \frac{1}{10}\mathbf{c}(3t) - \frac{1}{14}\mathbf{c}(4t) \\ & z = 1 \end{aligned} \quad (\text{IV.10})$$

$$\begin{aligned} \phi_2(t) : \quad & x = \frac{4}{5}\mathbf{s}^3(t) \\ & y = \frac{13}{20}\mathbf{c}(t) - \frac{1}{4}\mathbf{c}(2t) - \frac{1}{20}\mathbf{c}(3t) - \frac{1}{20}\mathbf{c}(4t) \\ & z = 1 \end{aligned} \quad (\text{IV.11})$$

Trajectory 3 is a parametric curve in three dimensions. Fig. IV.3(b) represents the trajectory in the workspace of the manipulator. The parametric equations of the trajectory is defined by Eq. (IV.10). $\phi_3(t)$ is the parametric equation for Trajectory 3 and is defined for $t \in [0, 20]$.

$$\phi_3(t) : \quad x = \mathbf{s}(t) \quad y = \mathbf{c}(t) \quad z = \frac{1}{20}t \quad (\text{IV.12})$$

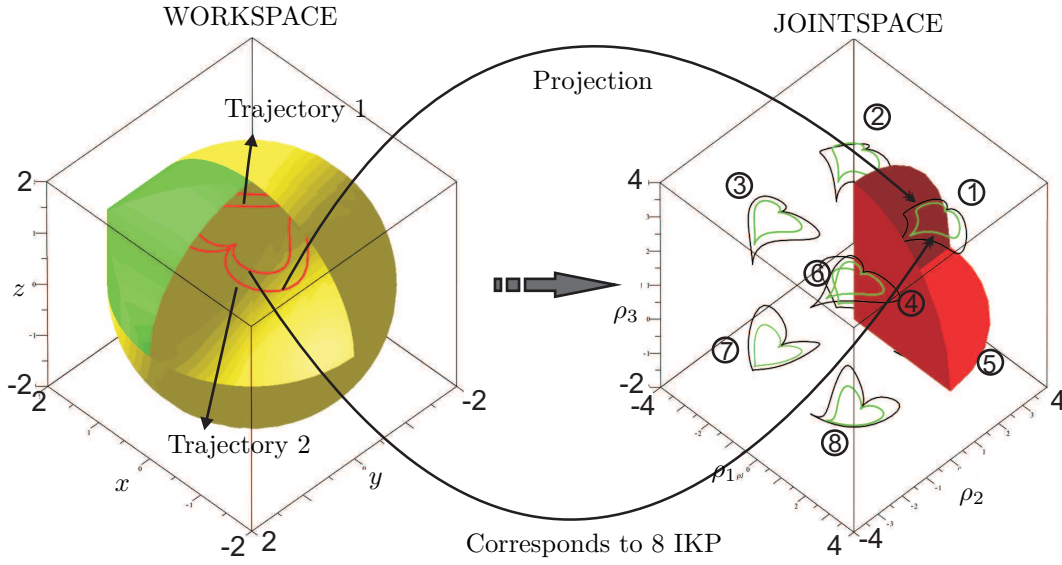


Figure IV.4 – A pictorial representation of the mapping of trajectories from workspace to joint space. Eight different pairs of trajectories, $\Upsilon_1(\rho, \mathbf{t})$ & $\Upsilon_2(\rho, \mathbf{t})$ in joint space are the image of corresponding $\phi_1(\mathbf{t})$ & $\phi_2(\mathbf{t})$. These eight different trajectories are associated with the eight working modes of the Orthoglide. Only one trajectory lies inside the joint space boundary due to the joint constraints.

In order to turn the system to an algebraic one, we add the following equations

$$\begin{aligned} \sin(t) &= \sin_t & \cos(t) &= \cos_t \\ \sin_t^2 + \cos_t^2 &= 1 \end{aligned} \quad (\text{IV.13})$$

and we remark that this change of variables does not introduce any spurious solutions since it is bijective ($t \in [-\pi, \pi]$).

IV.3.3 Projection in the joint space

$\phi_1(t)$, $\phi_2(t)$ and $\phi_3(t)$ are the trajectories which are defined in the workspace. To project these trajectories in the joint space, it is necessary to formulate a system of equations corresponding to each trajectory which also consists the kinematic equations of the manipulator. Ψ_1 , Ψ_2 and Ψ_3 are the corresponding systems of equations for the Trajectories 1, 2 and 3, respectively (see Eq. (V.10)).

$$\begin{aligned} \Psi_1(\mathbf{X}, \rho, \mathbf{t}) &= [\phi_1(\mathbf{t}) - \mathbf{X}, \mathbf{F}(\rho, \mathbf{X})] \\ \Psi_2(\mathbf{X}, \rho, \mathbf{t}) &= [\phi_2(\mathbf{t}) - \mathbf{X}, \mathbf{F}(\rho, \mathbf{X})] \\ \Psi_3(\mathbf{X}, \rho, \mathbf{t}) &= [\phi_3(\mathbf{t}) - \mathbf{X}, \mathbf{F}(\rho, \mathbf{X})] \end{aligned} \quad (\text{IV.14})$$

Each system of is projected in the joint space as $\Upsilon_1(\rho, \mathbf{t})$, $\Upsilon_2(\rho, \mathbf{t})$ and $\Upsilon_3(\rho, \mathbf{t})$ (see Eq. (V.11)). By solving $\Upsilon_1(\rho)$, $\Upsilon_2(\rho)$ and $\Upsilon_3(\rho)$ for ρ , we get the corresponding parametric equations for the trajectories in the joint space, as shown in Fig. IV.4 for the Trajectories 1&2 and in Fig. IV.5 for the Trajectory 3. Eqs. (IV.16) - (IV.18) are used to plot all

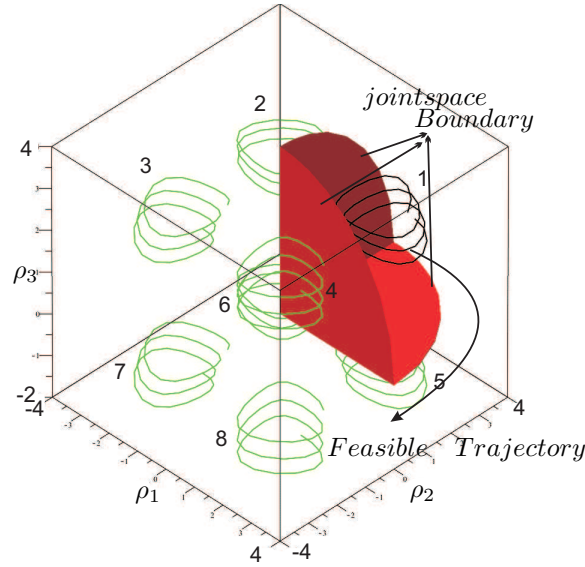


Figure IV.5 – Mapping of the Trajectory 3 from workspace to joint space. Eight different possible solutions of $\Upsilon_3(\rho, \mathbf{t})$ are marked in joint space which are the image of $\phi_3(\mathbf{t})$. There is only one feasible trajectory (marked as 1) lies inside the joint space boundary.

possible images of the trajectories in the joint space. All the computations are done without considering the joint limits. As the Orthoglide has eight working mode, there exists eight possible trajectories in the joint space (Fig. IV.4 and Fig. IV.5 for Trajectories 1&2 and Trajectory 3, respectively).

$$\begin{aligned}
 \Psi_1(\mathbf{X}, \rho, \mathbf{t}) &\mapsto \Upsilon_1(\rho, \mathbf{t}) & \forall t \in [-\pi, \pi] \\
 \Psi_2(\mathbf{X}, \rho, \mathbf{t}) &\mapsto \Upsilon_2(\rho, \mathbf{t}) & \forall t \in [-\pi, \pi] \\
 \Psi_3(\mathbf{X}, \rho, \mathbf{t}) &\mapsto \Upsilon_3(\rho, \mathbf{t}) & \forall t \in [0, 20]
 \end{aligned} \tag{IV.15}$$

Solving $\Upsilon_1(\rho, \mathbf{t})$ and $\Upsilon_2(\rho, \mathbf{t})$ for ρ gives eight possible solutions (Eq. (IV.17) for Trajectory 1 and Eq. (IV.18) for Trajectory 2), which inferred eight different possible images of Trajectory 1&2 in joint space. But from Fig. IV.4 one can see that only one trajectory lies inside the joint space boundary for the both trajectories in workspace.

$$\begin{aligned}
 \rho_1 &= \mathbf{s}(t) \pm \frac{1}{20} \sqrt{400\mathbf{s}^2(t) - t^2 + 1200} \\
 \rho_2 &= \mathbf{c}(t) \pm \frac{1}{20} \sqrt{400\mathbf{c}^2(t) - t^2 + 1200} \\
 \rho_3 &= \frac{1}{20}t \pm \sqrt{3}
 \end{aligned} \tag{IV.16}$$

Figure IV.5 shows all the possible images of Trajectory 3 in the joint space. Solving $\Upsilon_3(\rho, \mathbf{t})$ for ρ gives eight possible solutions which (Eq. (IV.16)). It can be seen in the Fig. IV.5 that only one trajectory (which is marked as 1), lies inside the joint space boundary.

$$\begin{aligned}
\rho_1 &= \frac{8}{7}\mathbf{s}^3(t) \pm \frac{1}{35}\sqrt{1996\mathbf{s}^6(t) - 400\mathbf{s}^8(t) + 560\mathbf{s}^6(t)\mathbf{c}(t) - 100\mathbf{s}^4(t)\mathbf{c}(t) - 2009\mathbf{s}^4(t) + \zeta_1} \\
\rho_2 &= -\frac{4}{7}\mathbf{c}^4(t) - \frac{2}{5}\mathbf{c}^3(t) - \frac{1}{7}\mathbf{c}^2(t) + \frac{43}{35}\mathbf{c}(t) + \frac{2}{7} \pm \frac{1}{7}\sqrt{64\mathbf{c}^6(t) - 192\mathbf{c}^4(t) + 192\mathbf{c}^2(t) + 83} \\
\rho_3 &= 1 \pm \frac{1}{35}\sqrt{3200 - 400\mathbf{c}^8(t) - 560\mathbf{c}^7(t) + 1204\mathbf{c}^6(t) + 1580\mathbf{c}^5(t) - 3221\mathbf{c}^4(t) + \zeta_2} \\
\zeta_1 &= 2190\mathbf{c}^3(t) - 1379\mathbf{c}^2(t) - 1320\mathbf{c}(t) + 3988 \\
\zeta_2 &= 710\mathbf{c}^3(t) + 3051\mathbf{c}^2(t) - 860\mathbf{c}(t)
\end{aligned} \tag{IV.17}$$

$$\begin{aligned}
\rho_1 &= \frac{4}{5}\mathbf{s}^3(t) \pm \frac{1}{10}\sqrt{76\mathbf{s}^6(t) - 16\mathbf{s}^8(t) + 16\mathbf{s}^6(t)\mathbf{c}(t) + 12\mathbf{s}^4(t)\mathbf{c}(t) - 85\mathbf{s}^4(t) + 96\mathbf{c}^3(t) - \zeta_3} \\
\rho_2 &= -\frac{2}{5}\mathbf{c}^4(t) - \frac{1}{5}\mathbf{c}^3(t) - \frac{1}{10}\mathbf{c}^2(t) + \frac{4}{5}\mathbf{c}(t) + \frac{1}{5} \pm \frac{1}{5}\sqrt{16\mathbf{c}^6(t) - 48\mathbf{c}^4(t) + 48\mathbf{c}^2(t) + 59} \\
\rho_3 &= 1 \pm \frac{1}{10}\sqrt{332 - 16\mathbf{c}^8(t) - 16\mathbf{c}^7(t) + 52\mathbf{c}^6(t) + 60\mathbf{c}^5(t) - 145\mathbf{c}^4(t) + 24\mathbf{c}^3(t) + \zeta_4} \\
\zeta_3 &= 66\mathbf{c}^2(t) - 60\mathbf{c}(t) + 321 \\
\zeta_4 &= 132\mathbf{c}^2(t) - 32\mathbf{c}(t)
\end{aligned} \tag{IV.18}$$

IV.3.4 Singularity analysis

To check if there exists any singular configuration between the two poses of the end-effector it is necessary to express the Jacobian of the manipulator as a function of the time or of some independent variable. The proposed algebraic method enables us to write the Jacobian of the manipulator as a function of the time and to check if its determinant vanishes between two poses.

By substituting the values of $\phi_1(t)$, $\phi_2(t)$ and $\phi_3(t)$ from Eq. (IV.10), Eq. (IV.11) and Eq. (IV.12) in $\mu(\mathbf{X})$, we will get $\mu(\phi_1(t))$, $\mu(\phi_2(t))$ and $\mu(\phi_3(t))$ as the determinant of the Jacobian for Trajectory 1, Trajectory 2 and Trajectory 3 respectively (Eq. (IV.19)). Due to the large expressions, the equations for $\mu_1(\mathbf{t})$, $\mu_2(\mathbf{t})$ and $\mu_3(\mathbf{t})$ are not presented but their roots define the singular configurations : Figures IV.8, IV.10 and IV.11 show the values of these functions when t varies.

$$\begin{aligned}
\mu_1(\mathbf{t}) &= \mu(\phi_1(\mathbf{t})) & \forall \mathbf{t} \in [-\pi, \pi] \\
\mu_2(\mathbf{t}) &= \mu(\phi_2(\mathbf{t})) & \forall \mathbf{t} \in [-\pi, \pi] \\
\mu_3(\mathbf{t}) &= \mu(\phi_3(\mathbf{t})) & \forall \mathbf{t} \in [0, \mathbf{20}]
\end{aligned} \tag{IV.19}$$

The number of solutions of Eq. (IV.19) gives the total number of singular points on the corresponding trajectory. The solutions for $\mu_1(\mathbf{t})$, $\mu_2(\mathbf{t})$ and $\mu_3(\mathbf{t})$ are shown in Eq. (IV.20). From Eq. (IV.20) we get that there exists four solutions for $\mu_1(\mathbf{t})$ and zero solutions for $\mu_2(\mathbf{t})$ and $\mu_3(\mathbf{t})$, which confirms the presence of singular points on Trajectory 1 whereas Trajectory 2&3 are singularity-free trajectories. Note that numerical approximations are given for readability, but, in practice, isolating intervals with rational bounds with arbitrary width can be computed (see section METHODOLOGY or [85, chapter 8] for more details) so that the result is certified.

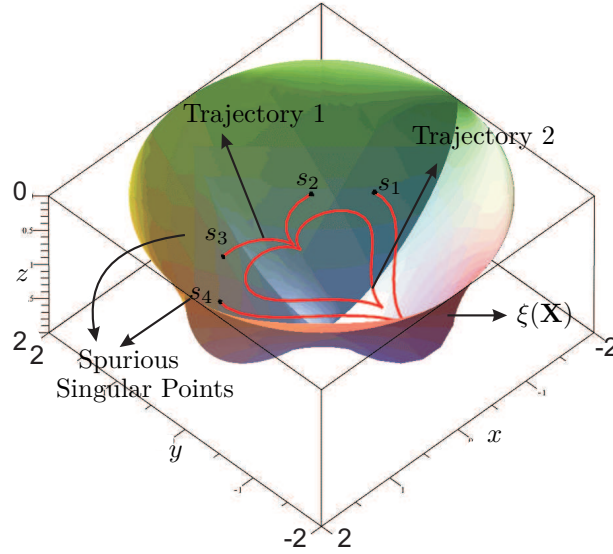


Figure IV.6 – Representation of the trajectories with singularity surface. Trajectory 1 cuts the parallel singularity surface $\xi(\mathbf{X})$ in four points s_1 , s_2 , s_3 and s_4 in the workspace of the Orthoglide. Also, it can be seen that Trajectory 2 lies inside $\xi(\mathbf{X})$ as $\mu_2(\mathbf{t}) \neq \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi]$.

$$\begin{aligned}
 \mathbf{t} &= [-1.51, -0.97, 0.97, 1.51] \leftarrow \mu_1(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi] \\
 \mathbf{t} &= [0.97, 1.51] \leftarrow \det(\mathbf{A}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi] \\
 \mathbf{t} &= [\quad] \leftarrow \mu_2(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi] \\
 \mathbf{t} &= [\quad] \leftarrow \mu_3(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [0, 20]
 \end{aligned} \tag{IV.20}$$

$\mu_1(\mathbf{t})$ is the determinant of the Jacobian corresponding to the Trajectory 1. By solving a zero-dimensional system of equations, it can be shown that Trajectory 1 cuts the singularity surface in four points s_1 , s_2 , s_3 and s_4 (see Fig. IV.8 and Eq. (IV.20)). The image of these points in workspace is computed by substituting the values of Eq. (IV.20) in Eq. (IV.10). Similarly, the image of these points in joint space is computed by substituting the values of Eq. (IV.20) in Eq. (IV.17). The obtained values are tabulated in Table IV.1. Figures IV.6 represents the Trajectory 1&2 along with singularity surface $\xi(\mathbf{X})$, also, all the singular points \mathbf{s}_1 , \mathbf{s}_2 , \mathbf{s}_3 and \mathbf{s}_4 are located on the singularity surface.

\mathbf{s}_1 and \mathbf{s}_2 are the spurious singular points which were introduced by the projection of $\det(\mathbf{A})$ in Cartesian space. Substituting $\mu_1(\mathbf{t})$ and $\phi_1(\mathbf{t})$ in $\det(\mathbf{A})$ gives two solutions (Eq. (IV.20)). These two solutions \mathbf{s}_3 and \mathbf{s}_4 are the real singular points on Trajectory 1. The variation of $\det(\mathbf{A})$ along Trajectory 1 is shown in Fig. IV.9.

The true singular postures \mathbf{s}_3 and \mathbf{s}_4 , belonging to the singularity surface $\xi(\mathbf{X})$, which is associated with the working mode used by the Orthoglide prototype (Fig. IV.7). $\mu_2(\mathbf{t})$ and $\mu_3(\mathbf{t})$ are the determinants of the Jacobians corresponding to the Trajectories 2&3 respectively.

The variation of $\mu_2(\mathbf{t})$ along Trajectory 2 for $t \in [-\pi, \pi]$ is shown in Fig. IV.10. Figure IV.11 represents the variation of $\mu_3(\mathbf{t})$ for $t \in [0, 20]$ along Trajectory 3. All the values

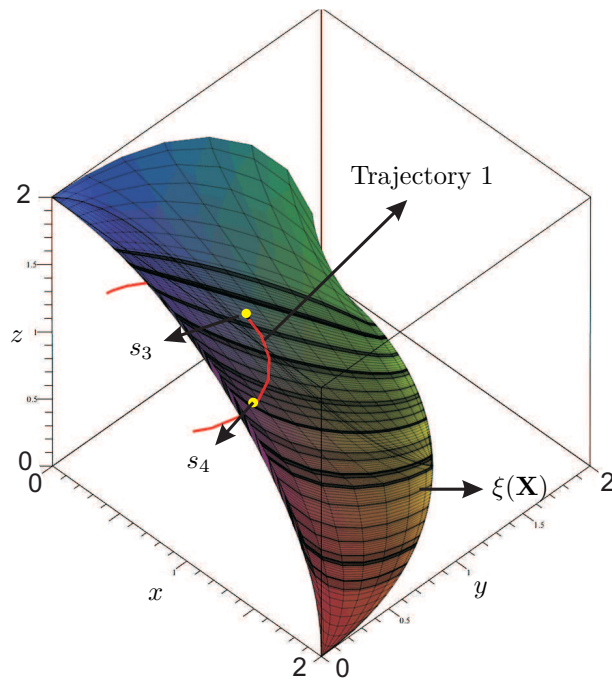


Figure IV.7 – Representation of the real singular points s_3 and s_4 along with the singularity surface $\xi(\mathbf{X})$, which is associated with one working mode.

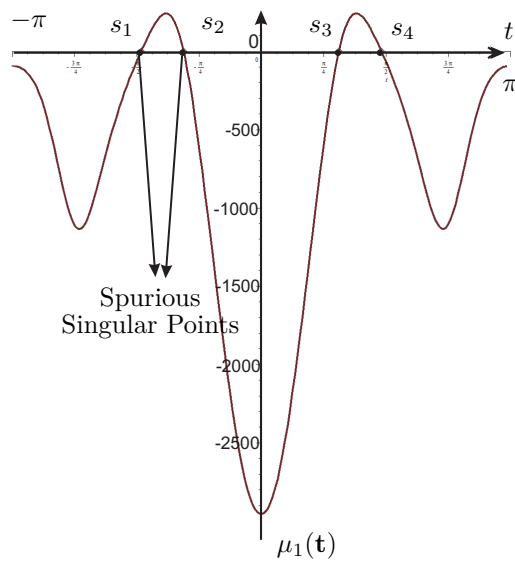
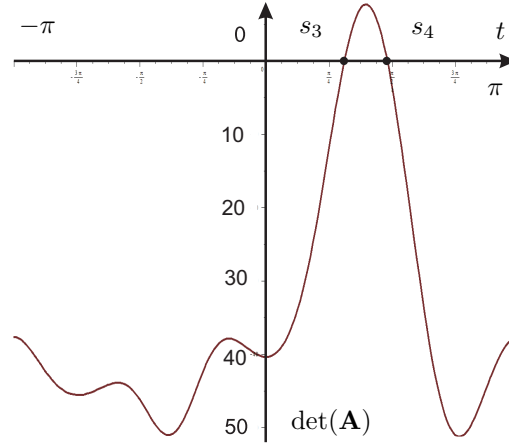


Figure IV.8 – Variation of $\mu_1(t)$ along Trajectory 1. s_1 , s_2 , s_3 and s_4 represents the four solutions of $\mu_1(t) = 0 \quad \forall t \in [-\pi, \pi]$.

Table IV.1 – Singular postures on Trajectory 1

Workspace				Joint space		
S	x	y	z	ρ_1	ρ_2	ρ_3
S_1	-1.13	0.35	1.00	0.55	1.66	2.60
S_2	-0.65	0.80	1.00	0.88	2.40	2.71
S_3	0.65	0.80	1.00	2.18	2.40	2.71
S_4	1.13	0.35	1.00	2.83	1.66	2.60

**Figure IV.9** – Variation of $\det(\mathbf{A})$ along the Trajectory 1. s_3 and s_4 represents the two solutions of $\det(\mathbf{A}) = 0 \quad \forall t \in [-\pi, \pi]$.

given in Table IV.1 are associated with the trajectories which is marked as 1 in Fig. IV.4. These values are obtained by solving a zero-dimensional system of polynomial equations using RootFinding:-Isolate so that the related values for \sin_t and \cos_t are certified.

In Table IV.1, \mathbf{s}_1 and \mathbf{s}_2 are the spurious singular points and \mathbf{s}_3 and \mathbf{s}_4 are the real singular points on the Trajectory 1. Trajectory 2&3 are the singularity-free trajectories.

IV.4 Parallel Manipulator with Several Actuation Modes

There are many types of reconfigurable parallel manipulators. This section deals with the determination of a control scheme for a reconfigurable parallel manipulator with different actuation schemes. It is well known that the kinetostatic performance of parallel manipulators may change drastically in the neighborhood of parallel singularities. Parallel mechanisms may have several solutions to their inverse kinematics problem, i.e. several working modes. It is usually difficult to get a large singularity-free workspace for a given working mode. Hence, one must plan the trajectory in such a way that there should be change of working mode to avoid the singularities. These problems can be solved by introducing actuation redundancy, which involves force control algorithms [87]. The kinetostatic performance of a planar parallel mechanism, NaVARo, with variable actuation modes is presented in [86, 88]. The homing, calibration and model-based predictive control for the 4RPR parallel architecture is shown in [89].

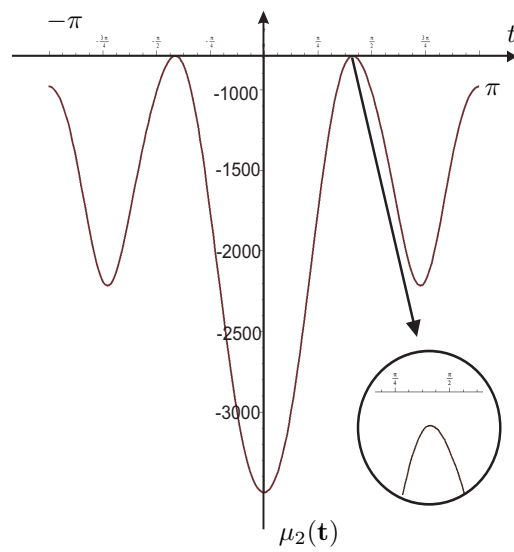


Figure IV.10 – Variation of $\mu_2(\mathbf{t})$ along the Trajectory 2. There are no solutions for $\mu_2(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi]$.

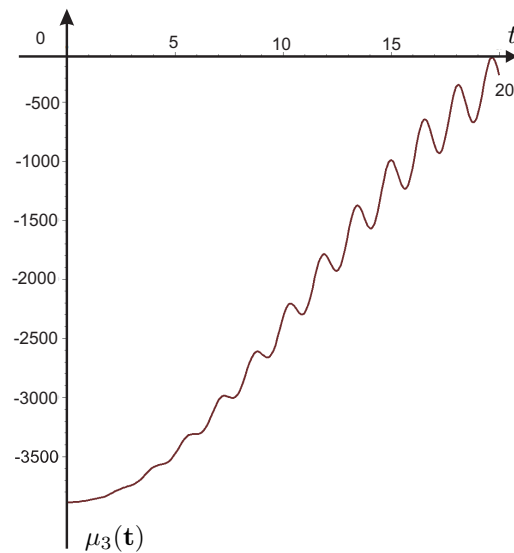


Figure IV.11 – Variation of $\mu_3(\mathbf{t})$ along the Trajectory 3. There are no solutions for $\mu_3(\mathbf{t}) = \mathbf{0} \quad \forall \mathbf{t} \in [-\pi, \pi]$.

In [90], a 3-CRU reconfigurable parallel manipulator is presented, which can work in two different modes, the first mode gives three degrees of freedom of pure translation motion and another of pure rotation. Moreover, it is shown that there is a possibility of switching between the two modes by simultaneous change of configuration of all the universal joints. The reconfiguration analysis of a 2-DOF 3-RRRR parallel mechanism is presented in [91], which has six operation modes, two spherical translational modes, two planar motion modes and two sphere-on-sphere rolling modes. A reconfigurable parallel mechanism with planar five bar metamorphic linkages is presented in [92]. A self-calibration method, which is based on leg-end distance errors and measurement residues for the three-legged modular reconfigurable parallel robots are presented in [93] and [95], respectively. In [96] a metamorphic parallel mechanism is presented which can switch its motion from a 3-dof translational motion to a 3-dof orientation motion. This paper presents the actuation modes of a variable actuation parallel manipulator, NaVARo. Besides, a framework is presented to follow any trajectory using its inverse geometric model only.

IV.4.1 Robot under study

The NaVARo robot was introduced in [86] and illustrated in Fig. IV.12. From the classical 3-RRR parallel robot, a parallelogram is added to each leg and connected to the first limb. A virtual model of NaVARo robot is shown in Fig. IV.13 which is used for analyzing the different aspects associated with the robot under study. Virtually, the actuator positions can change from the revolute joint located on A_i or B_i . This feature exists thanks to a clever transmission system made with two clutches. The NaVARo robot admits eight actuation modes if we assume that only one clutch can be activated in the same time per leg.



Figure IV.12 – The NaVARo robot

The dimensions were chosen so that the forward kinematics of the 3-RPR robot associated can be solved analytically. Thus, the points A_i and C_i form an equilateral triangle, respectively, and their sides has the dimension 0.7 m and 0.35 m, respectively. The lengths of the links A_iB_i and B_iC_i are 0.21 m for $i = 1, 2, 3$. Due to the parallelograms, the minimum and maximum distance between the points A_i and C_i is 0.039 m and 0.08 m, respectively.

IV.4.2 Transmission System

A transmission system has been developed and mounted in each leg of the NaVARo in order for the manipulator to be able to switch smoothly from one actuation mode to another along a prescribed trajectory. Figure IV.14 illustrates a CAD modeling of the transmission system of the NaVARo. This system can be seen as a double clutch system and contains: (i) a motor; (ii) a gearhead, (iii) a motor shaft, (iv) a main shaft (in cyan), (v) a base (in yellow), (vi) a housing (in purple) and (vii) two clutches (in brown). As a matter of fact, the two clutches 1 and 2 are electromagnetic brakes.

Each transmission system has four actuation schemes that are defined thereafter [98]:

- None of clutches 1 and 2 are active: The main shaft is free to move with respect to the housing and the base. In that case, none of the first two revolute joints of the corresponding legs are actuated, namely, angles θ_i and δ_i are passive, $i = 1, 2, 3$.
- Clutch 1 is active while Clutch 2 is not: The main shaft is fixed with respect to the base, i.e., the link A_iB_i is driven thanks to the rotation of the motor shaft. In that case, angle θ_i is active and angle δ_i is passive.

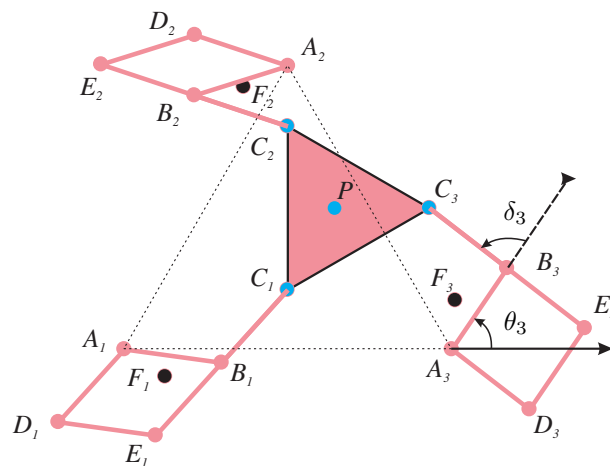


Figure IV.13 – The virtual model of the NaVARo robot

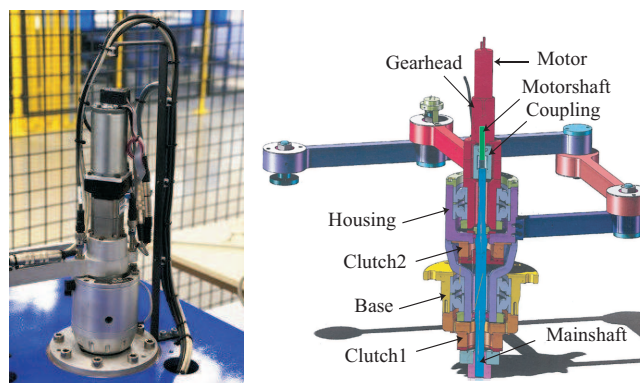


Figure IV.14 – The NaVARo transmission system

- Clutch 2 is active while Clutch 1 is not: The main shaft is attached to the housing, but is free to move with respect to the base. In that case, angle θ_i is passive and angle δ_i is active, $i = 1, 2, 3$.
- Both clutches 1 and 2: The blue shaft is attached to both the base and the housing. It means that the housing cannot move and link $A_i D_i$ is fixed. In that case, link $C_i E_i$ performs a circular translation with respect to point A_i . This actuation scheme amounts to an actuated II joint.

Only the second and third actuation schemes of each transmission system are used in the NaVARo prototype in order to keep the three degrees of freedom motion of the moving-platform and to avoid any actuation redundancy and under-actuation.

The fourth mode of actuation is used to lock the robot during the actuation mode changing or during the data transfer phases.

Table IV.2 – The eight actuation modes of the NaVARo

Actuation Mode Number	Driven Links	Active Links
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 B_1, A_2 B_2, A_3 B_3$	$\theta_1, \theta_2, \theta_3$
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 B_1, A_2 B_2, A_3 E_3$	$\theta_1, \theta_2, \delta_3$
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 B_1, A_2 E_2, A_3 B_3$	$\theta_1, \delta_2, \theta_3$
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 E_1, A_2 B_2, A_3 B_3$	$\delta_1, \theta_2, \alpha_3$
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 B_1, A_2 E_2, A_3 E_3$	$\theta_1, \delta_2, \delta_3$
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 E_1, A_2 E_2, A_3 B_3$	$\delta_1, \delta_2, \theta_3$
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 E_1, A_2 B_2, A_3 E_3$	$\delta_1, \theta_2, \delta_3$
$\underline{RRR}_1 - \underline{RRR}_2 - \underline{RRR}_3$	$A_1 E_1, A_2 E_2, A_3 E_3$	$\delta_1, \delta_2, \delta_3$

IV.4.3 Control Hardware and Software

The Control hardware of the NaVARo robot is a 1103 Dspace card [46] with a 450 Mhz Power PC. The actuator positions are required with a frequency equal to 6 kHz and a 200 Hz low pass filter is used to compute the actuator velocity. The robot motions are controlled thanks to a sub program working at 1 kHz. The state of the program is determined as a function of the set of actuators used.

The dynamic effect is added to the robot control scheme to improve the classical PID control scheme. The motor torques Γ_i are generated by removing the coupling between the legs as,

$$\Gamma_i = \mathbf{J} \left(\ddot{\theta} + \mathbf{K}_P (\theta_i^d - \theta_i) + \mathbf{K}_D (\dot{\theta}_i^d - \dot{\theta}_i) + \mathbf{K}_I \int_{t_0}^t (\theta_i^d - \theta_i) \right) \quad (\text{IV.21})$$

Where $J = 0.7k.m^2$, $\omega = 14 \text{ rad.s}$, $K_P = 3\omega^2$, $K_D = 3\omega$ and $K_I = \omega^3$.

IV.4.4 Sensor Placements

The NaVARo robot is equipped with three types of sensor. Incremental sensors are placed on the shaft of the motors. Analog angle sensors are placed on the pivots B_i and C_i to have an absolute measure of these angles. The sensors placed on the pivots B_i allow us to have

a measure of the distance $\|A_i C_i\|$ that is equivalent to consider a robot 3-RPR. Magnetic sensors located at F_i are used for the homing robot and at A_i to set the fine leg strokes to prevent the parallelogram is completely squashed or stretched. The location of the magnetic sensors will be described in the Homing Section.

IV.4.5 Graphic user interface

A graphic user interface (GUI) was developed in Matlab to make first the homing and after the motions (Fig. IV.15). When the start conditions are satisfied, the homing function can be performed (Section IV.5) and the movements of the robots are made in Cartesian mode (Section IV.6). The interface displays the value of the actuated and passive joints as well as the current actuation mode. It is also possible to release the servo-motors to move the robot by hand.

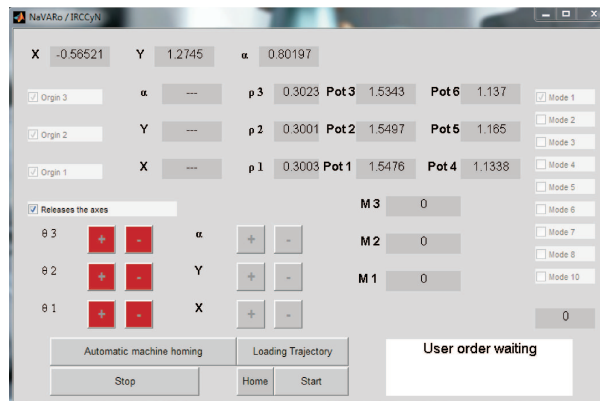


Figure IV.15 – The graphic user interface of the NaVARo robot

IV.5 Homing

IV.5.1 Introduction

The homing procedure permits the robot to reach a predefined known position by making a set of actions. The strategy used in our case is to manually place the robot in a given position, checking by sensors the absolute position, then start a movement to be accurately detect magnetic sensors. To understand the process, it is necessary to recall some properties of the robot.

IV.5.2 Direct kinematics of the 3-RPR

When the actuation mode 8 is activated, that is to say, a robot 3-RRR, has direct kinematic model equivalent to that of a robot 3-RPR where $r_i = \|A_i C_i\|$ for $i = 1, 2, 3$. The 3-RPR planar parallel robot with base and mobile platform admits only a maximum of four real solutions to the direct kinematic problem (DKP) [99], shown in Fig. IV.16(a).

The orientation of the mobile platform can be found by solving a second degree polyno-

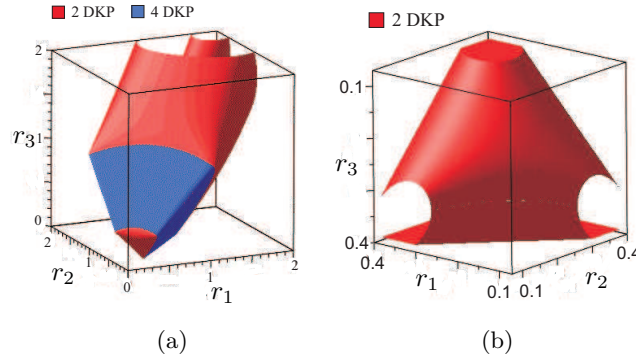


Figure IV.16 – Joint space of 3RPR parallel mechanism (a) without joint limits (b) with joint limits. The Red and Blue regions represent the two and four real solutions for the DKP, respectively.

mial equation in $\cos(\alpha)$:

$$\begin{aligned}
 & (78400 r_1^2 + 78400 r_2^2 + 78400 r_3^2 - 96040) \cos(\alpha) + \\
 & 160000 r_1^4 - 160000 r_1^2 r_2^2 - 160000 r_3^2 r_1^2 + 160000 r_2^4 - \\
 & 160000 r_3^2 r_2^2 + 160000 r_3^4 - 98000 r_1^2 - 98000 r_2^2 - \\
 & 98000 r_3^2 + 60025 + 38416 \cos(\alpha)^2 = 0
 \end{aligned} \tag{IV.22}$$

As this quadratic equation depends only on $\cos(\alpha)$, we have a maximum of four solutions. The position x and y can be directly defined as:

$$x = \frac{10 \left(6 (r_1^2 + r_3^2) - 3C\alpha (r_1^2 - r_3^2) - \sqrt{3}S\alpha (2r_2^2 - r_3^2 - r_1^2) \right)}{105 - 84C\alpha} \tag{IV.23}$$

$$y = \frac{10\sqrt{3} \left(2(r_1^2 + 2r_2^2 - r_3^2) - S\alpha\sqrt{3}(r_1^2 - r_3^2) - C\alpha(r_1^2 - 2r_2^2 + r_3^2) \right)}{105 - 84C\alpha} \tag{IV.24}$$

where $C\alpha = \cos(\alpha)$ and $S\alpha = \sin(\alpha)$. Due to the joint limits, there are only two real solutions to the direct kinematic problem which is shown in Fig. IV.16(b).

IV.5.3 Direct kinematics of the 3-RRR

Except for the actuation mode 8, the forward kinematics is general and there are up to 6 real solutions, or assembly modes. Solving this problem amounts to solving a polynomial of degree 6 using the method introduced by Gosselin [99]. This cannot be realized in our old DSpace card at the frequency of 1 kHz. However, it is possible to verify the current location by shifting the calculation on the computer controlling the robot. In our case, MATLAB computes all the solutions of the direct geometric model. However, only the absolute sensors on passive joints distinguish the current assembly mode. In all cases, the resolution is generic and comparable to the resolution of a forward kinematics of the robot 3-RPR is considering the length $\|A_i C_i\|$ or $\|B_i C_i\|$ as actuated articulation knowing the angle δ_i or θ_i respectively.

IV.5.4 Homing procedure

The home pose lies at the center of the workspace. The proximal link of the legs have to be in front of the magnetic sensor. By knowing the position of this segment is equivalent to study 3-RRR robot with the first motorized joint. To reach this configuration, we can move the robot in the joint space. We have defined a graphical interface for the displacement of each actuator by addition or subtraction of 5 degrees. At each movement, a trajectory based on a five-degree polynomial is used to have as a desired, zero speed and zero acceleration at the starting point and ending point. Since the algorithm is based on state, a single command can be run simultaneously. When the magnetic sensors are close to the proximal limb, a flag associated with each sensor becomes turned on.

If the robot reaches the limits of movement of parallelograms, security robot is activated and the power of the engine is turned off. It is necessary to move the mobile platform by external effort to depart from this position. For this set of input values, this robot admits two real solutions to the direct geometric model with the same position but two different orientations. As one can not distinguish them by the only position of the proximal, we use sensors placed on the second joint of each leg that has been previously calibrated. For $\theta_1 = -0.585$, $\theta_2 = -2.680$ and $\theta_3 = 1.508$, values for the position of the mobile platform x , y and α_i and the angles γ_i are

- $x = 0$, $y = 0$, $\alpha = 1.0471$ or $\alpha = 0.0414$
- $\delta_1 = \delta_2 = \delta_3 = 1.970$ or $\delta_1 = \delta_2 = \delta_3 = 1.005$

The two assembly modes for the NaVARo robot for the same position of the proximal limbs is shown in Fig. IV.17.

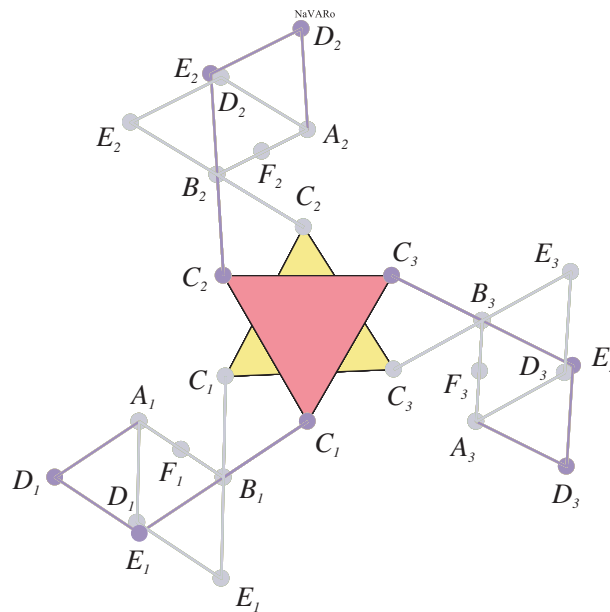


Figure IV.17 – First assembly mode of the NaVARo shown with pink colored mobile platform and the second assembly mode shown with yellow colored mobile platform for $\theta_1 = -0.585$, $\theta_2 = -2.680$ and $\theta_3 = 1.508$

The homing function is to move from offset each engine separately in the same direction and then return at a constant speed to the sensor. By identifying this position from the home position, we move the robot by joint offset. A flow chart of this algorithm is given in the Fig. IV.18. When homing is completed, the value encoder is initialized by setting an offset relative to their current values.

The state of the robot is 0 when no movement or command is active. Only the PID keeps the robot in its current position. When the state is not 0, no other command can be initiated. Only security-related commands can interrupt the program. When transferring information associated with a path, the movements are stopped and all clutches are active because this is a priority process. Movement generators are written in C language for generating movements according to the current posture and the end posture.

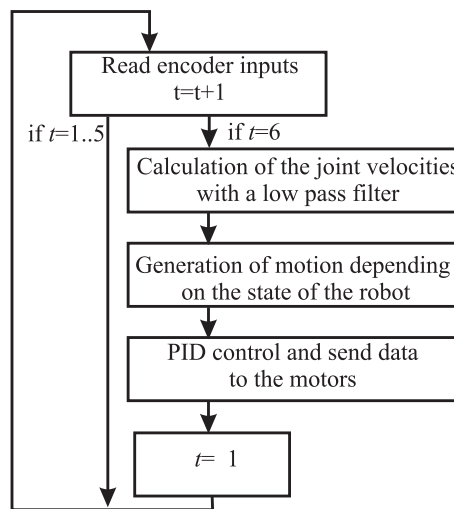


Figure IV.18 – Flow chart of the control loop

IV.6 Motions of the robot

To avoid solving the forward kinematics, when the boot process is complete, only Cartesian movements are possible. To change the actuation mode, it is necessary to know how to calculate the inverse kinematics for the first and second pivot joint of each leg. Similarly, to avoid singular configurations, it is necessary to evaluate the Jacobian matrices by building the position of all segments of the robot.

IV.6.1 Inverse kinematic model

For changing the actuation mode, the computation of general inverse kinematic mode is need to switch the value of the motor incremental sensor to the current actuated joint. In a first step, we calculate the position of C_i points depending on the position (x, y) and the orientation α of the mobile platform.

Knowing the current operating mode, the inverse geometric model calculates values of θ_i and δ_i .

IV.6.2 Jacobian matrices

As NaVARO robot allows only one working mode, knowing the position of the mobile platform and using the inverse kinematics, we can know the position of the proximal and distal of the three legs. The kinematic modeling was introduced in [88] for this robot. The following equation defines the velocity $\dot{\mathbf{p}}$ of the point P as a function of the three legs.

$$\dot{\mathbf{p}} = \dot{\theta}_i \mathbf{E}(\mathbf{b}_i - \mathbf{a}_i) + \dot{\delta}_i \mathbf{E}(\mathbf{c}_i - \mathbf{b}_i) + \dot{\alpha} \mathbf{E}(\mathbf{p} - \mathbf{c}_i) \quad (\text{IV.31})$$

where

$$\mathbf{E} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (\text{IV.32})$$

and \mathbf{a}_i , \mathbf{b}_i , \mathbf{c}_i and \mathbf{p} are the position vectors of points A_i , B_i , C_i and P and $\dot{\theta}_i$, $\dot{\delta}_i$ and $\dot{\alpha}$ are the joint rates θ_i , δ_i and α as it is shown in Fig. IV.19.

A generic equation was obtained in [88] to define the Jacobian matrix of NaVARO as

$$\mathbf{A} \mathbf{t} = \mathbf{B} \dot{\mathbf{q}} \quad \text{with} \quad \mathbf{t} = [\dot{\mathbf{p}} \quad \dot{\alpha}]^T \quad \dot{\mathbf{q}} = [\dot{q}_1 \dot{q}_2 \dot{q}_3]^T \quad (\text{IV.33})$$

where $\dot{\mathbf{q}}$ is the joint rate vector which depends on the actuator that is to say the actuated joint, ie $\dot{\theta}_i$ or $\dot{\delta}_i$. The direct and inverse Jacobian matrices are this defined

$$\mathbf{A} = \begin{bmatrix} (\mathbf{c}_1 - \mathbf{h}_1)^T & -(\mathbf{c}_1 - \mathbf{h}_1)^T \mathbf{E} (\mathbf{p} - \mathbf{c}_1) \\ (\mathbf{c}_2 - \mathbf{h}_2)^T & -(\mathbf{c}_2 - \mathbf{h}_2)^T \mathbf{E} (\mathbf{p} - \mathbf{c}_2) \\ (\mathbf{c}_3 - \mathbf{h}_3)^T & -(\mathbf{c}_3 - \mathbf{h}_3)^T \mathbf{E} (\mathbf{p} - \mathbf{c}_3) \end{bmatrix} \quad (\text{IV.34})$$

$$\mathbf{B} = \text{diag} \left[(\mathbf{c}_i - \mathbf{b}_i)^T \mathbf{E} (\mathbf{b}_i - \mathbf{a}_i) \right] \quad \text{for} \quad i = 1, 2, 3 \quad (\text{IV.35})$$

where $\mathbf{h}_i = \mathbf{a}_i$ if the first joint of the leg is actuated and $\mathbf{h}_i = \mathbf{b}_i$ in the other case. To normalize the Jacobian matrix, we use as the characteristic value, the radius r of the mobile platform. Thus the matrix \mathbf{A} becomes

$$\mathbf{A} = \begin{bmatrix} (\mathbf{c}_1 - \mathbf{h}_1)^T & -(\mathbf{c}_1 - \mathbf{h}_1)^T \mathbf{E} (\mathbf{p} - \mathbf{c}_1)/r \\ (\mathbf{c}_2 - \mathbf{h}_2)^T & -(\mathbf{c}_2 - \mathbf{h}_2)^T \mathbf{E} (\mathbf{p} - \mathbf{c}_2)/r \\ (\mathbf{c}_3 - \mathbf{h}_3)^T & -(\mathbf{c}_3 - \mathbf{h}_3)^T \mathbf{E} (\mathbf{p} - \mathbf{c}_3)/r \end{bmatrix} \quad (\text{IV.36})$$

Then, we compute the Jacobian matrix and its condition number as

$$\mathbf{J} = \mathbf{B}^{-1} \mathbf{A} \quad (\text{IV.37})$$

$$\kappa(\mathbf{J}) = \frac{1}{m} \sqrt{\text{tr}(\mathbf{J}^T \mathbf{J}) \text{tr}(\mathbf{J}^T \mathbf{J})^{-1}} \quad (\text{IV.38})$$

The minimum value of κ^{-1} is limited to 0.1 to avoid the singular configurations during the motions.

IV.6.3 Actuation mode changing

Because of the design of our system, the actuation mode changing can not be done only when the robot is stopped. When a clutch is deactivated, the connection between the motor and a fixed reference of the robot is eliminated. Each path ends at zero speed and we expect the robot position is stable and the convergence of the PID is done. The transition from one actuation mode to another by activating all the clutches and then releasing the unnecessary clutch to the desired actuation mode. When all clutches are activated, the robot becomes over-constrained and no movement of the platform is possible. It is in this configuration that the data encoders switches it from one mode to the other. The method is to calculate the inverse kinematics introduced previously. First, we store the values of each actuated joint in a vector according to the current actuation mode ϵ_{old} , then we do the same with the new mode of actuation ϵ_{new} . The difference between ϵ_{old} and ϵ_{new} sets the offset that is added on the encoder positions.

IV.6.4 Motions of the robot

When the homing is completed, it is not possible to move the robot in a joint mode. However, the motions of the robot can be realized with two other modes. The first is with the GUI where one can realize translational or rotational movements by pressing buttons up / down of each axis. The conditioning of the Jacobian matrix is evaluated before each motion or before each actuation mode changing to prohibit the motions of the robot near singular configurations. With the second mode, we can load movements that have been generated by another program written in Matlab. We have, for example, a program that reads a G code and generate a file with the position, speed and acceleration sets as well as an integer to define the associated actuation mode. It is thus possible to make a singular assembly mode changing trajectory if the conditions defined in [97] are validated. The video of a path example is given in [100].

IV.7 Conclusions

This chapter reports the use of algebraic methods to check the feasibility of given trajectories in the workspace. The general method uses the projection of the polynomial equations associated with the trajectories in the joint space using a Gröbner based elimination strategy. There is a significant change in the shape of workspace and the singularity surfaces due to the joint constraints. The proposed method enables us to project the joint constraints in the workspace of the manipulator, which further helps to analyse the projection of the singularities in the workspace with the joint constraints. Such computations ensure the existence of a singular configuration between two poses of the end-effector unlike other numerical or discretization methods. This paper highlights that the singularity analysis should be done in the cross product of the workspace and joint space for parallel robots with several assembly and working modes. The single analysis of the singularities in a projection space can introduce spurious singularities. In fact, the algebraic tools does not allow to distinguish the parallel singularities according to the working mode. With NaVARo

robot, a variable actuation robot, it is possible to introduce all the problems related to the operating principle as the loss of connection between the encoder and the frames of the robot. A start procedure allows to identify the current assembly mode of the robot by using magnetic sensors and absolute encoders for low accuracy. The movements sensed by the magnetic sensors at low speed and always in the same direction can guarantee the reproducibility of the procedure. A framework is presented to realize the trajectory that does not use the direct geometric model, which uses only the inverse geometric model. The redundant sensors ensure that the current method of assembly for the command is the one used by the robot. This allows to perform assembly mode change paths and confirms that the movement has been achieved.

V

Prospective Work: Evaluation of the Accuracy

Usually, the accuracy of parallel manipulators depends on the architecture of the robot, the design parameters, the trajectory planning and the location of the path in the workspace. This chapter reports the influence of static and dynamic parameters in computing the error in the pose associated with the trajectory planning made and analyzed with the Orthoglide 5-axis. An error model is proposed based on the joint parameters (velocity and acceleration) and experimental data coming from the Orthoglide 5-axis. Newton and Gröbner based elimination methods are used to project the joint error in the workspace to check the accuracy/error in the Cartesian space. For the analysis, five similar trajectories with different locations inside the workspace are defined using fifth order polynomial equation for the trajectory planning. It is shown that the accuracy of the robot depends on the location of the path as well as the starting and the ending posture of the manipulator due to the acceleration parameters.

V.1 Introduction

The accuracy to attain the exact pose for the parallel manipulator for a given trajectory depends on the static and dynamic parameters associated with the manipulator. Due to the better dynamic properties, high load-carrying capacity, high accuracy and stiffness, closed loop mechanisms are best suited for the medical robotics, high-precision and machine tool design applications. Number of links and passive joints in the closed loop mechanism reduces the accuracy of the manipulator. There are different factors which affects the accuracy of the manipulator, some of them are geometrical deviation of the machine parts during their assembly, mechanism motion, elastic deformation of the links and joints due to the force and thermal expansion [32, 33]. There are several article exists on the effect of manufacturing tolerances on the accuracy of the parallel manipulators [34]. In [35], a forward and inverse error bound analysis is presented to find the error bound in the pose of the end effector for a Gough-Stewart platform when the joint error bounds are given and vice versa. The sensitivity analysis for a three degrees-of-freedom translational parallel kinematic machine with orthogonal linear joints is reported in [36]. They have used linkage kinematic analysis

and differential vector method to study the influence of the length variation on the pose of the end-effector.

One of the highly addressed problem associated with the end-effector pose error is the manipulator stiffness, which defines the positioning error due to the external loading while executing a specific task by the manipulator. A non-linear stiffness model for the manipulators with the passive joints is presented in [37]. Pashkevich et al [38] proposed a novel calibration approach for the Orthoglide based on the observations of the manipulator leg parallelism. A numerical procedure presented in [39], which is used to compute the pose error due to clearances and elastic deformations along a pick-and-place trajectory of the 5R planar parallel manipulator. In [40], an error prediction model is proposed for overconstrained and nonoverconstrained parallel manipulators and also presents the influence of the joint tolerance on the pose error of the manipulators. A larger number of contributions in the literature is available on the influence of the statics and robot Jacobian than the influence of dynamics on computing the error in the pose of the parallel manipulator [41, 42, 94]. A methodology is presented in [43] to project the trajectories in the joint space using Gröbner based elimination methods. This paper mainly focuses on the estimation of error in the pose of end effector due to the joint errors produced by the PID control loop. In this article, the proposed error model which is based on the dynamic properties (joint velocities and acceleration) of the Orthoglide helps in estimating the error in the Cartesian workspace.

The outline of this chapter is as follows. We first introduced the direct and inverse kinematic model of the Orthoglide with parallel singularities. In the next section we have defined five different trajectories for the Orthoglide 5-axis, which has been developed at IRCCyN. We then proposed the joint error model based on the joint velocities, acceleration and joint errors which are obtained with the Orthoglide 5-axis during motions. In the later section we have presented the estimation of the error of the end-effector pose with Newton and Gröbner based elimination methods.

V.2 Architecture & Kinematics: Orthoglide 5-AXIS

The Orthoglide 5-axis is a hybrid parallel kinematics machine, which consists of a 3-DOF translational manipulator (Orthoglide 3-axis) and a 2-DOF parallel spherical manipulator (the Agile Eye 2-axis) as two of the main components [44]. The manipulator under study is a semi industrial prototype of the Orthoglide 5-axis, which is designed and developed at IRCCyN for the robotic team and manufactured by Symetrie [45], as shown in Fig. V.1.

Orthoglide 5-axis uses 1103 DSPACE card as a control hardware with a 933 MHz PowerPC. The trajectory planning is done using Matlab and an optical fiber is used to send the data to DSPACE card [46]. Two different types of motors are used to actuate the spherical and translational assembly. Two Harmonic Drives FFA-20-80-H-E2048 are used to actuate 2-DOF parallel spherical manipulator, which has 3.27 rad/s as the maximum angular velocity and 270 rad/s² as the maximum angular acceleration. Orthoglide 3-axis uses three Parvex Brushless NX430EAF coupled with Kinetic TDU 200 ball screws as actuators which

has a maximum linear velocity is equal to 1.2 m/s and maximum acceleration of 13 m/s². The actuator positions are sampled at 9 KHz and actuator velocities are computed using 200 Hz low pass filter. The gravity effect is reduced using pneumatic compensator which is mounted along the vertical axis. A constant pressure in the chamber of the cylinders allows to apply an upward force similar to the mass of the mobile platform and part of the mass of parallelograms. This pressure is defined by cutting the servo so that the vertical axis does not move downwardly or upwardly.

Equation (V.1) represents the constraint equations of the Orthoglide 3-axis, which defines the motion of three orthogonal linear actuators with $l = 310mm$, the actuated variables $\boldsymbol{\rho} = [\rho_1, \rho_2, \rho_3]$ and the pose variables $\mathbf{X} = [x, y, z]$:

$$\begin{aligned} F(\boldsymbol{\rho}, \mathbf{X}) : (x - \rho_1)^2 + y^2 + z^2 &= l^2 \\ x^2 + (y - \rho_2)^2 + z^2 &= l^2 \\ x^2 + y^2 + (z - \rho_3)^2 &= l^2 \end{aligned} \quad (\text{V.1})$$

$\|\mathbf{A}_i \mathbf{B}_i\|$ is equal to ρ_i , which represents the prismatic joint variables whereas \mathbf{X} represents the position vector of the tool center point as it is shown in the Fig. V.2. In the following sub sections direct and inverse kinematics model are derived for the Orthoglide 3-axis.

V.2.1 Direct Kinematics Model:

The problem of determining the pose of the end-effector of the manipulator for a given value of joint parameters is termed as a direct kinematic problem (DKP). With an algebraic model, there are chances of getting several solutions for DKP i.e the manipulator can be assembled in the different ways for a set of actuator values. The number of solutions for the DKP is referred as the assembly modes of the robot. There exist two real solutions for DKP or two assembly modes for the Orthoglide. Equation (V.2) represents the direct kinematic model for the Orthoglide 3-axis.

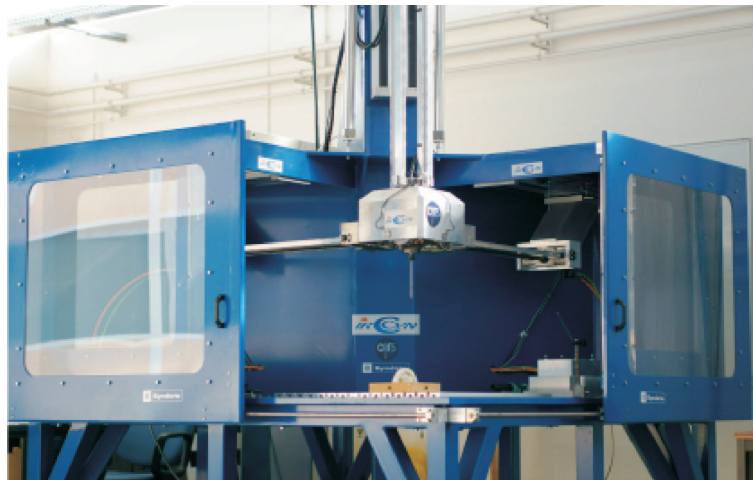


Figure V.1 – Semi industrial prototype of the Orthoglide 5-Axis

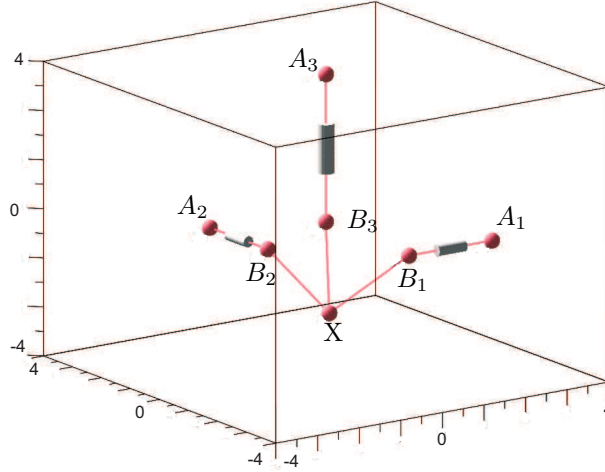


Figure V.2 – Simplified architecture of the Orthoglide for the simulation and the analysis.

$$\beta(\boldsymbol{\rho}) : \begin{aligned} x &= \frac{1}{2\rho_1} \frac{\rho_1^4 \rho_2^2 + \rho_1^4 \rho_3^2 - \sqrt{\mathbf{N}} \rho_3}{\mathbf{M}} \\ y &= \frac{1}{2\rho_2} \frac{\rho_1^4 \rho_2^2 + \rho_2^4 \rho_3^2 - \sqrt{\mathbf{N}} \rho_3}{\mathbf{M}} \\ z &= \frac{1}{2\rho_3} \frac{\rho_1^4 \rho_3^2 + \rho_2^4 \rho_3^2 - \sqrt{\mathbf{N}} \rho_3}{\mathbf{M}} \end{aligned}$$

$$\text{where } \mathbf{M} = \rho_1^2 \rho_2^2 + \rho_1^2 \rho_3^2 + \rho_2^2 \rho_3^2$$

$$\mathbf{N} = -\rho_1^6 \rho_2^4 - \rho_1^6 \rho_2^2 \rho_3^2 - \rho_1^4 \rho_2^6 - 2\rho_1^4 \rho_2^4 \rho_3^2 - \rho_1^4 \rho_2^2 \rho_3^4 - \rho_1^2 \rho_2^6 \rho_3^2 - \rho_1^2 \rho_2^4 \rho_3^4 + 384400(\rho_1^4 \rho_2^4 + \rho_1^4 \rho_2^2 \rho_3^2 + \rho_1^2 \rho_2^4 \rho_3^2) \quad (\text{V.2})$$

Solving Eq. (V.1) for \mathbf{X} gives two solutions corresponding to the two assembly modes of the Orthoglide. Equation (V.2) represents only one solution which corresponds to the required assembly modes. $\beta(\boldsymbol{\rho})$, the direct kinematic model is used to compute the projection of the joint errors in the workspace of the Orthoglide [38].

V.2.2 Inverse Kinematics Model:

Similarly the problem of determining the joint actuators value for a given pose of a end-effector is termed as a inverse kinematic problem (IKP). The number of solutions for the IKP is referred as the working modes of the robot [47]. There exist eight real solutions for IKP or the working modes for the Orthoglide 3-axis that can be computed using the Eq. (V.3).

$$\begin{aligned}
\gamma(\mathbf{X}) : \rho_1 &= x \pm \sqrt{-y^2 - z^2 + 96100} \\
\rho_2 &= y \pm \sqrt{-x^2 - z^2 + 96100} \\
\rho_3 &= z \pm \sqrt{-x^2 - y^2 + 96100}
\end{aligned} \tag{V.3}$$

V.2.3 Parallel Singularities

Singularities play an important role while defining the trajectory for a specific task. The error in the poses of the end-effector gradually increases when it approaches singular configuration. Differentiating Eq. (V.1) with respect to time leads to the velocity model:

$$\mathbf{A}\dot{\mathbf{X}} + \mathbf{B}\dot{\boldsymbol{\rho}} = \mathbf{0} \quad \text{where} \quad \mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{X}}, \quad \mathbf{B} = \frac{\partial \mathbf{F}}{\partial \boldsymbol{\rho}} \tag{V.4}$$

The matrices \mathbf{A} and \mathbf{B} are respectively the direct-kinematics and the inverse-kinematics Jacobian matrices of the manipulator. These matrices are used for characterizing different kinds of singularities. The parallel singularities occur whenever $\det(\mathbf{A}) = 0$, i.e, the mapping from tool velocity space to the joint velocity space is ill-conditioned and the serial singularities occur whenever $\det(\mathbf{B}) = 0$. The parallel singular configurations are located inside the workspace. They are particularly undesirable because the manipulator cannot resist to any forces and its control is lost. Parallel singularities occur when the target point and the center of prismatic joints are lie in the same plane or when all three links parallel to each other.

$$\begin{aligned}
\det(\mathbf{A}) &= -8\rho_1\rho_2\rho_3 + 8\rho_1\rho_2z + 8\rho_1\rho_3y + 8\rho_2\rho_3x \\
\det(\mathbf{A}) &\mapsto \varepsilon(\boldsymbol{\rho}) \quad \det(\mathbf{A}) \mapsto \xi(\mathbf{X}) \\
\varepsilon(\boldsymbol{\rho}) &= \rho_1^4\rho_2^2 + \rho_1^4\rho_3^2 + \rho_1^2\rho_2^4 + 2\rho_1^2\rho_2^2\rho_3^2 + \rho_1^2\rho_3^4 + \\
&\quad \rho_2^4\rho_3^2 + \rho_2^2\rho_3^4 - 16\rho_1^2\rho_2^2 - 16\rho_1^2\rho_3^2 - 16\rho_2^2\rho_3^2
\end{aligned} \tag{V.5}$$

In Eq. (V.5), $\det(\mathbf{A})$ is the parallel singularity of the Orthoglide and $\xi(\mathbf{X})$ is the projection of $\det(\mathbf{A})$ in the cartesian space. $\varepsilon(\boldsymbol{\rho})$ is the projection of parallel singularity $\det(\mathbf{A})$ in the joint space of the Orthoglide. The mathematical expression for $\xi(\mathbf{X})$ is not mentioned in Eq. (V.5) due to the lack of space. Figure V.3 shows the projection $\xi(\mathbf{X})$ in the workspace of the Orthoglide. The Gröbner basis and cylindrical algebraic decomposition (CAD) algorithms are used to compute and plot the workspace and parallel singularities of the Orthoglide [48, 49]. The degree of this surface is 18 and it represents the singularities associated with the eight working modes.

V.3 Trajectory Planning

The most simple and suitable technique for planning the trajectory for a specific task is always to define the path in the workspace $\mathbf{X} = [x, y, z]$ and control loop in the joint space

$\rho = [\rho_1, \rho_2, \rho_3]$ of the manipulator. Equation (V.6) represents the torque control structure for the actuators motion [44]. As it can be observed from Eq. (V.6), the torque Γ_i also depends on the dynamics of the manipulator which further enhances the classical PID control scheme.

$$\Gamma_i = M \left(\ddot{\rho}_i + K_P(\rho_i^d - \rho_i) + K_I \int_{t_0}^t (\rho_i^d - \rho_i) + K_D(\dot{\rho}_i^d - \dot{\rho}_i) \right)$$

for $i = 1, 2, 3,$ (V.6)

$$\text{where } K_P = 19200 \text{rad}^2/\text{s}^2, \quad K_I = 512000 \text{rad}^3/\text{s}^3$$

$$K_D = 240 \text{rad}/\text{s}, \quad M = 91.6278 \text{ Kg}$$

For the study five similar circular trajectories $\phi_1(t)$, $\phi_2(t)$, $\phi_3(t)$, $\phi_4(t)$ and $\phi_5(t)$ are defined but with the different center locations c_1 , c_2 , c_3 , c_4 and c_5 , respectively, inside the Orthoglide's workspace. These trajectories can be seen in Fig. V.3 which are labeled as 1, 2..5 with workspace and parallel singularity surfaces. The trajectories $\phi_i(t): [\tau_{x_i}, \tau_{y_i}, \tau_{z_i}]$ in Eq. (V.8) are defined in parametric form in the Cartesian space using fifth order polynomial equation with different center locations which is shown in Eq. (V.7).

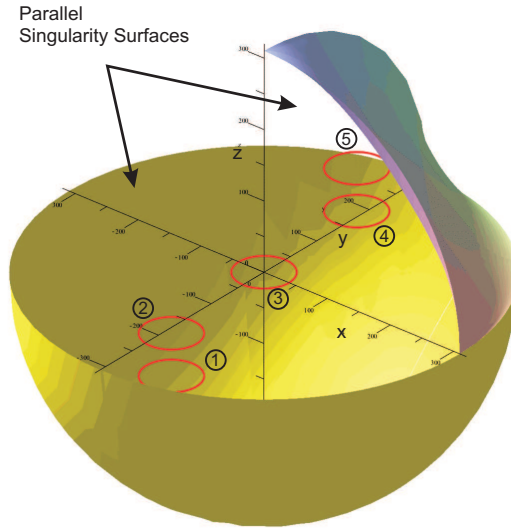


Figure V.3 – Location of the trajectories $\phi_i(t)$ along with the parallel singularity surfaces $\xi(\mathbf{X})$. CAD algorithm is used to plot these surfaces for a single working mode.

The execution time is set to one seconds for all the trajectories. The trajectories for Orthoglide 5-axis are defined such that α and β are equal to zero i.e, there exists only translational motion on xy plane with $\dot{x}_{max} = \dot{y}_{max} = 0.45 \text{m}/\text{s}$ and $\ddot{x}_{max} = \ddot{y}_{max} = 5 \text{m}/\text{s}^2$.

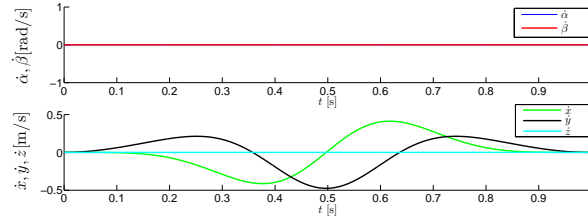


Figure V.4 – Cartesian velocities along the trajectories $\phi_i(t)$

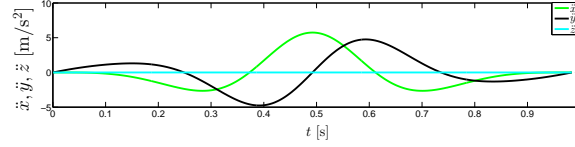


Figure V.5 – Cartesian acceleration along the trajectories $\phi_i(t)$

$$\begin{aligned}
 c_1 &: [c_{x_1}, c_{y_1}, c_{z_1}] \rightarrow [-80, -80, -140] \\
 c_2 &: [c_{x_2}, c_{y_2}, c_{z_2}] \rightarrow [-80, -80, -80] \\
 c_3 &: [c_{x_3}, c_{y_3}, c_{z_3}] \rightarrow [00, 00, 00] \\
 c_4 &: [c_{x_4}, c_{y_4}, c_{z_4}] \rightarrow [80, 80, 80] \\
 c_5 &: [c_{x_5}, c_{y_5}, c_{z_5}] \rightarrow [80, 80, 140]
 \end{aligned} \tag{V.7}$$

Differentiating Eq. (V.8) with respect to time gives us Cartesian velocities τ_{x_i} , τ_{y_i} and τ_{z_i} along x , y and z direction respectively, which is shown in Eq. (V.9). Similarly, Cartesian accelerations can be obtained by differentiating Cartesian velocities i.e, Eq. (V.9) with respect to time.

$$\begin{aligned}
 \phi_i(t) : \tau_{x_i} &= c_{x_i} + 40 \sin(2t^3(6t^2 - 15t + 10)\pi) - x \\
 \tau_{y_i} &= c_{y_i} + 40 \cos(2t^3(6t^2 - 15t + 10)\pi) - y \\
 \tau_{z_i} &= c_{z_i} - z \quad \forall t \in [0, 1]
 \end{aligned} \tag{V.8}$$

Figure V.4 and V.5 show the plot of Cartesian velocities and accelerations, respectively, along the trajectory $\phi_i(t)$. All the trajectories have the same Cartesian velocities and accelerations.

To project these trajectories in the joint space, it is necessary to formulate a system of equations corresponding to each trajectory which also consists the kinematic equations of the manipulator.

$$\begin{aligned}
 \dot{\tau}_{x_i} &= 2400t^2\pi(t^2 - 2t + 1) \cos(2t^3(6t^2 - 15t + 10)\pi) \\
 \dot{\tau}_{y_i} &= -2400t^2\pi(t^2 - 2t + 1) \sin(2t^3(6t^2 - 15t + 10)\pi) \\
 \dot{\tau}_{z_i} &= 0 \quad \forall t \in [0, 1]
 \end{aligned} \tag{V.9}$$

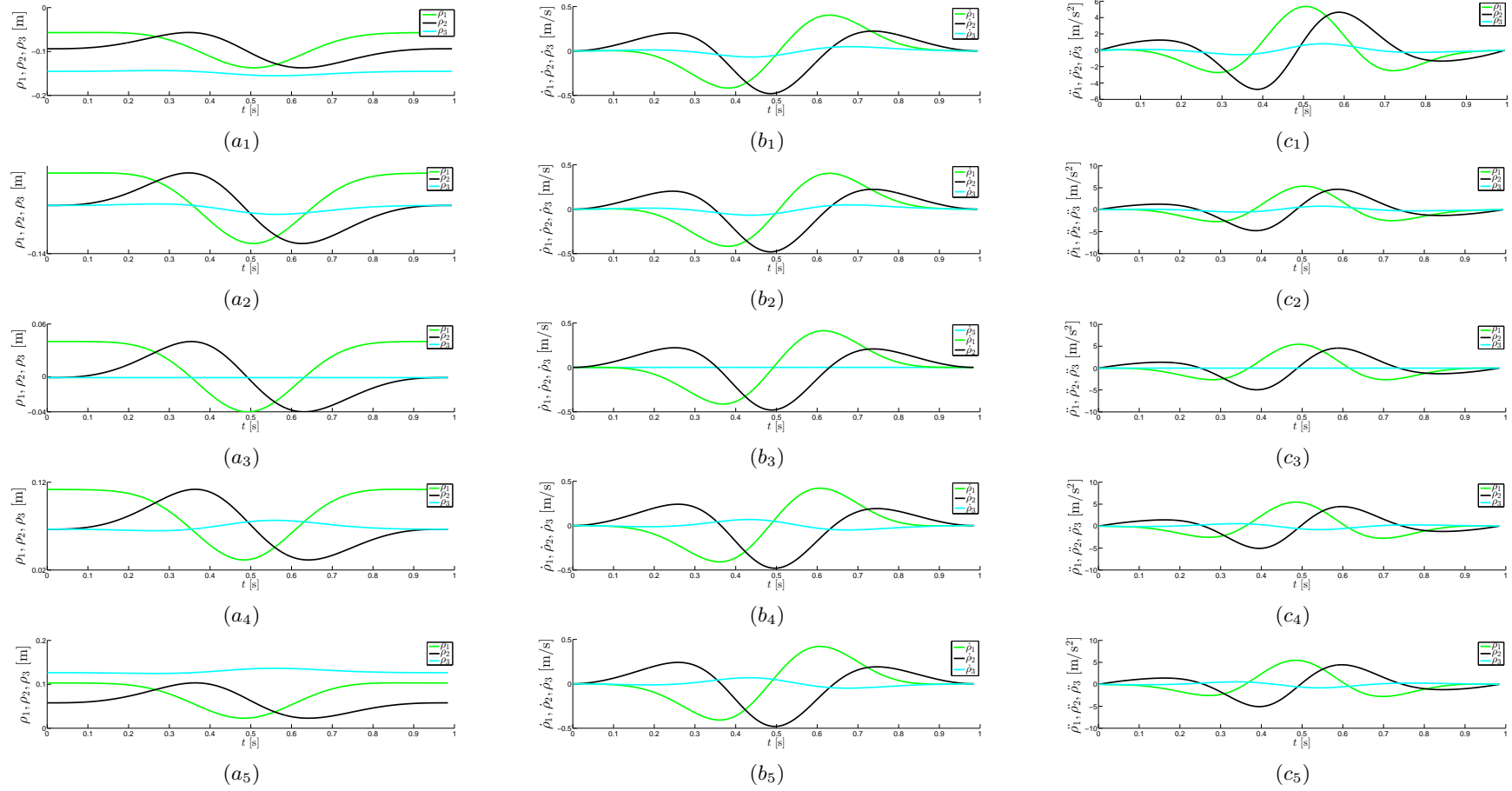


Figure V.6 – Joint parameters values along trajectories $\phi_1(t), \phi_2(t), \phi_3(t), \phi_4(t), \phi_5(t)$ Joint positions (a_i) Joint velocities (b_i) Joint accelerations (c_i), where $i = 1, 2, 3, 4, 5$ which represents different trajectories

$\Psi_1, \Psi_2, \Psi_3, \Psi_4$ and Ψ_5 are the corresponding systems of equations for the Trajectories 1, 2, 3, 4, and 5, respectively (see Eq. (V.10)).

$$\begin{aligned}
\Psi_1(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &= [[\tau_{x_1}, \tau_{y_1}, \tau_{z_1}], F(\boldsymbol{\rho}, \mathbf{X})] \\
\Psi_2(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &= [[\tau_{x_2}, \tau_{y_2}, \tau_{z_2}], F(\boldsymbol{\rho}, \mathbf{X})] \\
\Psi_3(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &= [[\tau_{x_3}, \tau_{y_3}, \tau_{z_3}], F(\boldsymbol{\rho}, \mathbf{X})] \\
\Psi_4(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &= [[\tau_{x_4}, \tau_{y_4}, \tau_{z_4}], F(\boldsymbol{\rho}, \mathbf{X})] \\
\Psi_5(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &= [[\tau_{x_5}, \tau_{y_5}, \tau_{z_5}], F(\boldsymbol{\rho}, \mathbf{X})]
\end{aligned} \tag{V.10}$$

Each system of equation is projected in the joint space as $\Upsilon_1(\boldsymbol{\rho}, \mathbf{t})$, $\Upsilon_2(\boldsymbol{\rho}, \mathbf{t})$ and $\Upsilon_3(\boldsymbol{\rho}, \mathbf{t})$ (see Eq. (V.11)). By solving $\Upsilon_1(\boldsymbol{\rho})$, $\Upsilon_2(\boldsymbol{\rho})$ and $\Upsilon_3(\boldsymbol{\rho})$ for $\boldsymbol{\rho}$, we get the corresponding parametric equations for the trajectories in the joint space. There are two different ways to compute these projections. The first method uses Gröbner based elimination algorithms and the second method is to directly substitute the $[\tau_{x_1}, \tau_{y_1}, \tau_{z_1}]$ in inverse kinematic model (Equation (V.3)).

$$\begin{aligned}
\Psi_1(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &\mapsto \Upsilon_1(\boldsymbol{\rho}, \mathbf{t}) & \forall t \in [0, 1] \\
\Psi_2(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &\mapsto \Upsilon_2(\boldsymbol{\rho}, \mathbf{t}) & \forall t \in [0, 1] \\
\Psi_3(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &\mapsto \Upsilon_3(\boldsymbol{\rho}, \mathbf{t}) & \forall t \in [0, 1] \\
\Psi_4(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &\mapsto \Upsilon_4(\boldsymbol{\rho}, \mathbf{t}) & \forall t \in [0, 1] \\
\Psi_5(\mathbf{X}, \boldsymbol{\rho}, \mathbf{t}) &\mapsto \Upsilon_5(\boldsymbol{\rho}, \mathbf{t}) & \forall t \in [0, 1]
\end{aligned} \tag{V.11}$$

Solving $\Upsilon_1(\boldsymbol{\rho}, \mathbf{t})$, $\Upsilon_2(\boldsymbol{\rho}, \mathbf{t})$, $\Upsilon_3(\boldsymbol{\rho}, \mathbf{t})$, $\Upsilon_4(\boldsymbol{\rho}, \mathbf{t})$ and $\Upsilon_5(\boldsymbol{\rho}, \mathbf{t})$ for $\boldsymbol{\rho}$ gives eight possible solutions for trajectories $\phi_i(t)$, but only one feasible solution exists which lies inside the joint space, which corresponds to each trajectories. The first column in the Figure V.6, i.e, Figure V.6(a_i) represents the joint actuator values along the trajectory ϕ_i , whereas second and third column, i.e, Figure V.6(b_i) & (c_i) represents the joint velocities and accelerations, respectively.

V.4 Error Analysis

During the execution of any given trajectory by the manipulator, there always exist some differences between the desired actuator values and the actual actuator values at any given instance of time. These differences are recorded by sensors which are generally attached with the actuators. Figure V.7 shows the recorded values of the joint errors while executing the trajectories by the Orthoglide 5-axis for the translational part.

The different parameters associated with the recorded joint errors $\Delta\rho_i$ along the the trajectories $\phi_i(t)$ in Fig. V.7 such as maximum, minimum and mean absolute values are presented in Table V.1. The maximum error in ρ_1 is more [697 μm] for the trajectory T_5 , whereas less [584.8 μm] for the trajectory T_1 . The mean absolute error in ρ_1 is the maximum [171.2 μm] for the trajectory T_5 and minimum [125.8 μm] for the trajectory

Table V.1 – The Minimum, Maximum and mean absolute values of joint errors along the trajectories

Errors (μm) \rightarrow	$\Delta\rho_1$			$\Delta\rho_2$			$\Delta\rho_3$		
Trajectories \downarrow	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
$T_1 : [-80, -80, -140]$	-280.6	584.8	125.8	-639.1	784.0	212.1	-401.4	558.0	151.0
$T_2 : [-80, -80, -80]$	-312.7	626.6	130.6	-665.2	777.7	220.2	-424.5	529.3	159.5
$T_3 : [00, 00, 00]$	-272.4	663.9	149.0	-615.1	666.1	204.2	-020.2	020.2	011.5
$T_4 : [80, 80, 80]$	-312.6	685.9	161.2	-636.7	566.6	135.8	-572.9	410.1	167.0
$T_5 : [80, 80, 140]$	-314.5	697.0	171.2	-630.9	565.0	138.6	-565.5	427.2	167.7

T_1 . The maximum error in ρ_2 is more [784 μm] for the trajectory T_1 , whereas less value [565 μm] for the trajectory T_5 . The mean absolute error in ρ_2 is the maximum [220.2 μm] for the trajectory T_2 and minimum [135.8 μm] for the trajectory T_4 . The maximum error in ρ_3 is more [558 μm] for the trajectory T_1 , whereas less [020.2 μm] for the trajectory T_1 . The mean absolute error in ρ_3 is the maximum [167.7 μm] for the trajectory T_5 and minimum [011.5 μm] for the trajectory T_3 .

There may exist several factors which affect these joints errors. It has also been observed that with the change of starting pose, accuracy of the manipulator also changes for a specified task. By comparing the data for different trajectories from Table V.1, it can be inferred that by changing the location of the trajectories inside the workspace of the Orthoglide, there is a significant change in the joint errors. It is difficult to choose the best suitable location for a trajectory based on the joint errors because we should evaluate their images in the workspace.

V.5 Error Model

One of the essential steps in computing the projection of the joint errors in the workspace is to define a joint error model. Our proposed model is based on experimental data and the dynamics, i.e, joint velocities and accelerations of the Orthoglide 5-axis.

$$\Delta\rho_i^m = k_{1i} + k_{2i}\dot{\rho}_i + k_{3i}\ddot{\rho}_i \quad (V.12)$$

Equation (V.12) represents the proposed error model in which k_{1i} , k_{2i} and k_{3i} have

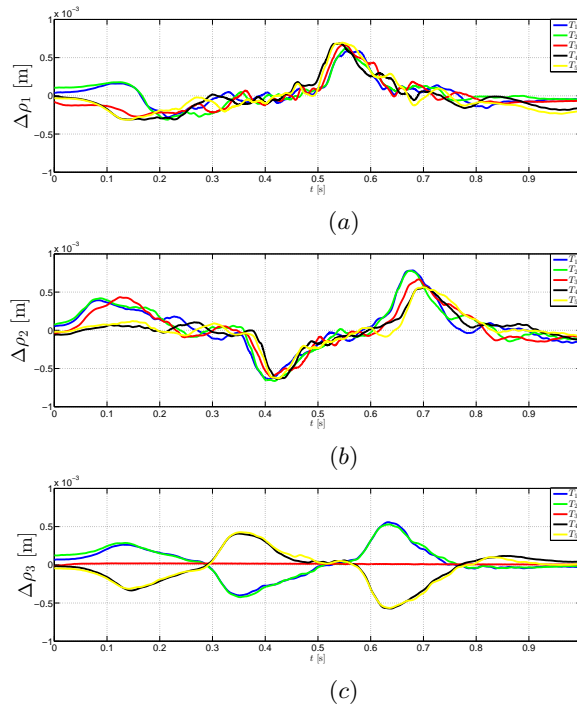


Figure V.7 – Joint errors in $\Delta\rho_1$ (a) $\Delta\rho_2$ (b) and $\Delta\rho_3$ (c) along the trajectories $\phi_i(t)$, Experimental data from the Orthoglide 5-axis

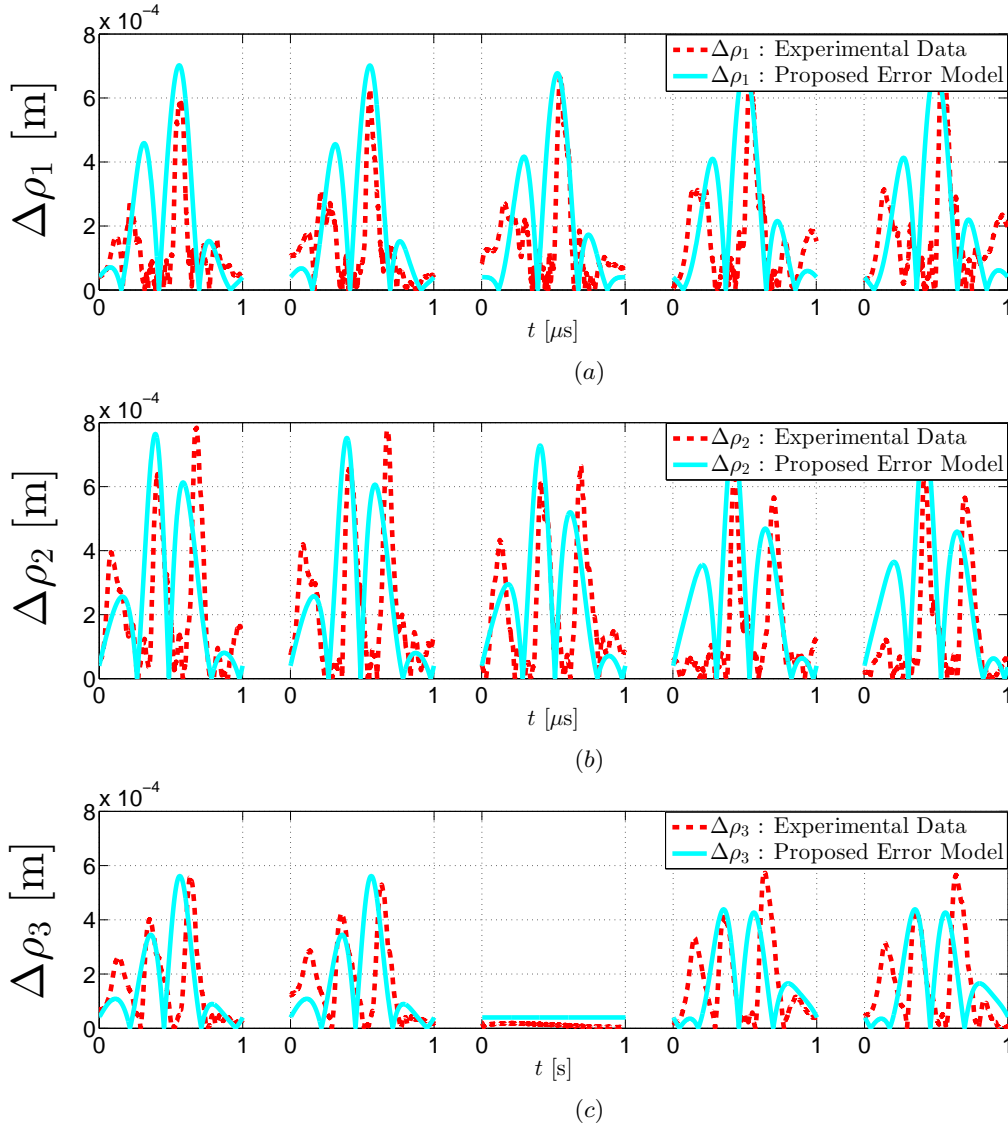


Figure V.8 – Joint error $\Delta\rho_i$ and proposed error model $\Delta\rho_i^m$ value along trajectory $\phi_1(t)$: $[-80, -80, -140]$, $\phi_2(t)$: $[-80, -80, -80]$, $\phi_3(t)$: $[00, 00, 00]$, $\phi_4(t)$: $[80, 80, 140]$, $\phi_5(t)$: $[80, 80, 140]$ $\Delta\rho_1$ (a) $\Delta\rho_2$ (b) $\Delta\rho_3$ (c), where $i = 1, 2, 3, 4, 5$ which represents different trajectories

Table V.2 – Maximum and mean absolute values of Cartesian position errors

Errors (mm)→	x		y		z	
Trajectories ↓	Max	Avg	Max	Avg	Max	Avg
$T_1 : [-80, -80, -140]$	0.90	0.34	1.10	0.31	0.90	0.25
$T_2 : [-80, -80, -80]$	0.80	0.30	0.90	0.28	0.70	0.21
$T_3 : [00, 00, 00]$	0.70	0.25	0.70	0.22	0.10	0.05
$T_4 : [80, 80, 80]$	1.00	0.40	0.80	0.38	0.80	0.34
$T_5 : [80, 80, 140]$	2.10	0.79	2.20	0.77	1.90	0.63

constant values and represent the static error, the friction and the dynamic effect and $\dot{\rho}_i$ and $\ddot{\rho}_i$ are the joint velocity and acceleration, respectively for $i = 1, 2, 3$. The modeled error $\Delta\rho_i^m$ and the experimental data $\Delta\rho_i$ from the manipulator for joint errors are compared to get the best suited values for k_{1i} , k_{2i} and k_{3i} which minimizing the least-squares error by using the Fit function of Maple [50].

$$\begin{aligned}
k_{11} &= \frac{1}{250000} & k_{21} &= \frac{1}{2000000} & k_{31} &= \frac{1}{8000000} \\
k_{12} &= \frac{1}{250000} & k_{22} &= \frac{1}{2000000} & k_{32} &= \frac{33}{250000000} \\
k_{13} &= \frac{1}{250000} & k_{23} &= \frac{1}{2000000} & k_{33} &= \frac{121}{500000000}
\end{aligned} \tag{V.13}$$

The computed values k_{1i} , k_{2i} and k_{3i} are shown in Eq. (V.12), by substituting these values in Eq. (V.12), we will get Eq. (V.14), which represents the proposed error model for the Orthoglide. More optimized algorithm can be used to compute the more appropriate values for the constants. We can notice that only the acceleration term is different for the z axis while for the x and y -axis, they are very similar. This means that the influence of gravity forces is not completely compensated with the pneumatic cylinders arranged in parallel to this axis or that they slow down the movement of the axis due to the compression of air in the circuit.

The black dotted line in the Fig. V.8 represents the joint errors which are obtained from the Orthoglide while executing the trajectory. The red line in the Fig. V.8 is the modeled error, which further will be used to compute the image of the joint errors in the workspace of the manipulator.

$$\begin{aligned}
\Delta\rho_1^m &= \frac{1}{250000} + \frac{1}{2000000}\dot{\rho}_1 + \frac{1}{8000000}\ddot{\rho}_1 \\
\Delta\rho_2^m &= \frac{1}{250000} + \frac{1}{2000000}\dot{\rho}_2 + \frac{33}{250000000}\ddot{\rho}_2 \\
\Delta\rho_3^m &= \frac{1}{250000} + \frac{1}{2000000}\dot{\rho}_3 + \frac{121}{500000000}\ddot{\rho}_3
\end{aligned} \tag{V.14}$$

V.6 Projection of Joint Errors in the Workspace

Newton method is used to compute the image of joint errors in the workspace of the manipulator. By differentiating Eq. (V.3), i.e, inverse kinematic equations and substituting

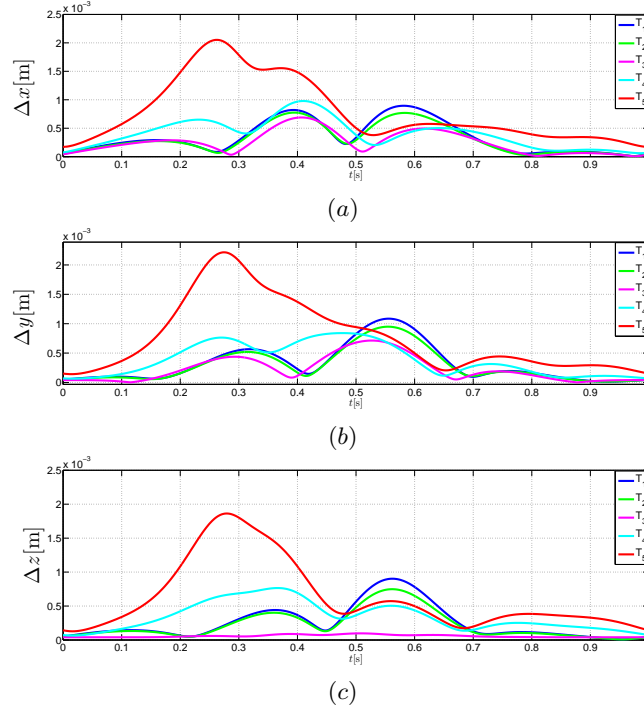


Figure V.9 – Cartesian errors in \mathbf{x} (a) \mathbf{y} (b) and \mathbf{z} (c) along the trajectories $\phi_i(t)$, Image of the joint errors in the workspace of the manipulator

the values of proposed error model $\Delta\rho_i^m$, i.e $e_1 = \Delta\rho_1^m$, $e_2 = \Delta\rho_2^m$ and $e_3 = \Delta\rho_3^m$ in Eq. (V.15), we will get the image of joint errors Δx , Δy and Δz in the workspace of the manipulator.

$$\begin{aligned}
 \Delta x^2 &= \left(\frac{\partial x}{\partial \rho_1}\right)^2 e_1^2 + \left(\frac{\partial x}{\partial \rho_2}\right)^2 e_2^2 + \left(\frac{\partial x}{\partial \rho_3}\right)^2 e_3^2 \\
 \Delta y^2 &= \left(\frac{\partial y}{\partial \rho_1}\right)^2 e_1^2 + \left(\frac{\partial y}{\partial \rho_2}\right)^2 e_2^2 + \left(\frac{\partial y}{\partial \rho_3}\right)^2 e_3^2 \\
 \Delta z^2 &= \left(\frac{\partial z}{\partial \rho_1}\right)^2 e_1^2 + \left(\frac{\partial z}{\partial \rho_2}\right)^2 e_2^2 + \left(\frac{\partial z}{\partial \rho_3}\right)^2 e_3^2
 \end{aligned} \tag{V.15}$$

The different parameters associated with the projected joint errors in the workspace Δx , Δy and Δz along the trajectories $\phi_i(t)$ in Fig. V.9 are presented in Table V.2. The maximum error in \mathbf{x} is more [2.1 mm] for the trajectory T_5 , whereas less [0.70 mm] for the trajectory T_3 . The mean absolute error in \mathbf{x} is the maximum [0.79 mm] for the trajectory T_5 and minimum [0.25 mm] for the trajectory T_3 . The maximum error in \mathbf{y} is more [2.20 mm] for the trajectory T_5 , whereas less [0.70 mm] for the trajectory T_3 . The mean absolute error in \mathbf{y} is the maximum [0.77 mm] for the trajectory T_2 and minimum [0.22 mm] for the trajectory T_3 . The maximum error in \mathbf{z} is more [1.90 mm] for the trajectory T_5 , whereas less [0.10 mm] for the trajectory T_3 . The mean absolute error in \mathbf{z} is the maximum [0.63 mm] for the trajectory T_5 and minimum [0.05 mm] for the trajectory T_3 .

Analysis: Influence of Location on accuracy

The five similar trajectories, labeled T_i where $i \in [1, 2, \dots, 5]$, are defined at different locations in the workspace of the Orthoglide 5-axis. The main idea of the experiment is to study the influence of location of the trajectory on the accuracy of the manipulator. The two trajectories, T_1 and T_5 , are located near the singularity surface and third trajectory, T_3 , closest to the isotropic posture. The other two trajectories, T_2 and T_4 , are in between these trajectories as shown in Fig. V.3. Experimental data from the sensor proves that there is a significant change in the joint errors during the execution of these trajectories. It is difficult to choose the best suitable location for a trajectory based on joint errors because we should evaluate their images in the workspace. The proposed error model, which is based on the joint velocities and acceleration enables us to project the joints error in the workspace more accurately unlike other model based on Jacobian or condition number. By analyzing these projected errors in the workspace, We have found out that the trajectory closer to the singularity surface, T_5 admitted the maximum error and is approximately twice than the trajectory closest to the isotropic posture, T_3 . The other reason for these differences in the accuracies may be the fact that the actuator motion depends on the location of the corresponding trajectory, while executing trajectory T_3 there exists only two actuator motion unlike in the case of execution of other trajectories.

V.7 Influence of Starting positions on the joints errors

The accuracy of the parallel manipulators depends on the architecture of the robot, the design parameters, the trajectory planning, the location of the path in the workspace and the starting position of the manipulator. From the previous section, we can depict that the trajectory closer to the singularity surface admitted the maximum error and the trajectory closest to the isotropic posture the minimum error. The proposed error model which is based on the dynamic properties of the Orthoglide helps in estimating the error in the Cartesian space. In this section, the influence of starting position of the manipulator on the phase error of the end-effector is shown. The accuracy of the path in the Cartesian workspace is studied for five similar circular trajectories at different locations and with different starting positions (shown in Fig. V.10) for each trajectory.

V.7.1 Joints errors for different starting positions

The difference between the desired actuator values and the actual actuator values are recorded by the sensors which are generally attached with the actuators. Figure V.11 presents the recorded values of the joint errors while executing the trajectories with different starting positions by the Orthoglide 5-axis. There are four different starting positions [$T_i : [0]$ (in blue), $T_i : [\pi/2]$ (in green), $T_i : [\pi]$ (in red), $T_i : [3\pi/2]$ (in black)], where $i = 1, 2, 3, 4, 5$, which represents different trajectories positions, chosen for each defined trajectory. The different parameters associated with the recorded joint errors $\Delta\rho_i$ along the trajectories $\phi_i(t)$ for different starting positions in Fig. V.11 such as maximum, minimum and mean absolute values are presented in Table V.3.

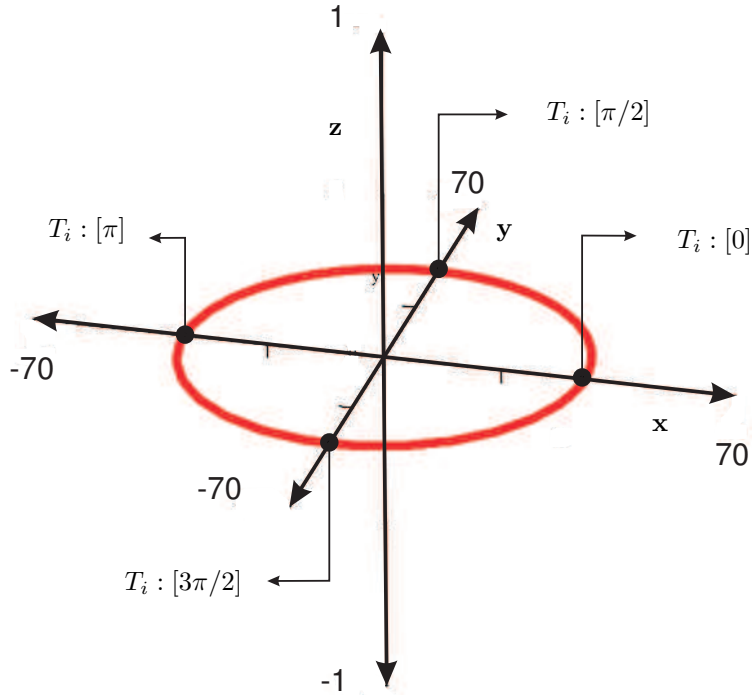


Figure V.10 – Different starting points on the trajectory to analyse the influence on the accuracy of the Orthoglide.

From the previous results, it can be inferred that the trajectory closer to the singularity surface admitted the maximum error and the trajectory closest to the isotropic posture the minimum error. Further, it can be observed by analyzing the experimental data, which is tabulated in Table V.3 that there is a significant change in the accuracy by changing the starting position of the manipulator.

Trajectory 1: T_1

The maximum error in ρ_1 is more [565.9 μm] for the starting position $T_1 : [3\pi/2]$, whereas less [387.1 μm] for the trajectory $T_1 : [\pi]$. The mean absolute error in ρ_1 is the maximum [158.5 μm] for the trajectory $T_1 : [3\pi/2]$ and minimum [121.2 μm] for the trajectory $T_1 : [\pi]$. The maximum error in ρ_2 is more [897.8 μm] for the trajectory $T_1 : [\pi/2]$, whereas less value [304.7 μm] for the trajectory $T_1 : [3\pi/2]$. The mean absolute error in ρ_2 is the maximum [185.3 μm] for the trajectory $T_1 : [\pi]$ and minimum [136.2 μm] for the trajectory $T_1 : [0]$. The maximum error in ρ_3 is more [524.6 μm] for the trajectory $T_1 : [\pi/2]$, whereas less [390.5 μm] for the trajectory $T_1 : [3\pi/2]$. The mean absolute error in ρ_3 is the maximum [151.4 μm] for the trajectory $T_1 : [3\pi/2]$ and minimum [112.9 μm] for the trajectory $T_1 : [0]$.

Trajectory 2: T_2

The maximum error in ρ_1 is more [626.6 μm] for the starting position $T_2 : [3\pi/2]$, whereas less [582.5 μm] for the trajectory $T_2 : [\pi]$. The mean absolute error in ρ_1 is the maximum [175 μm] for the trajectory $T_2 : [\pi]$ and minimum [125.5 μm] for the trajectory $T_2 : [0]$. The

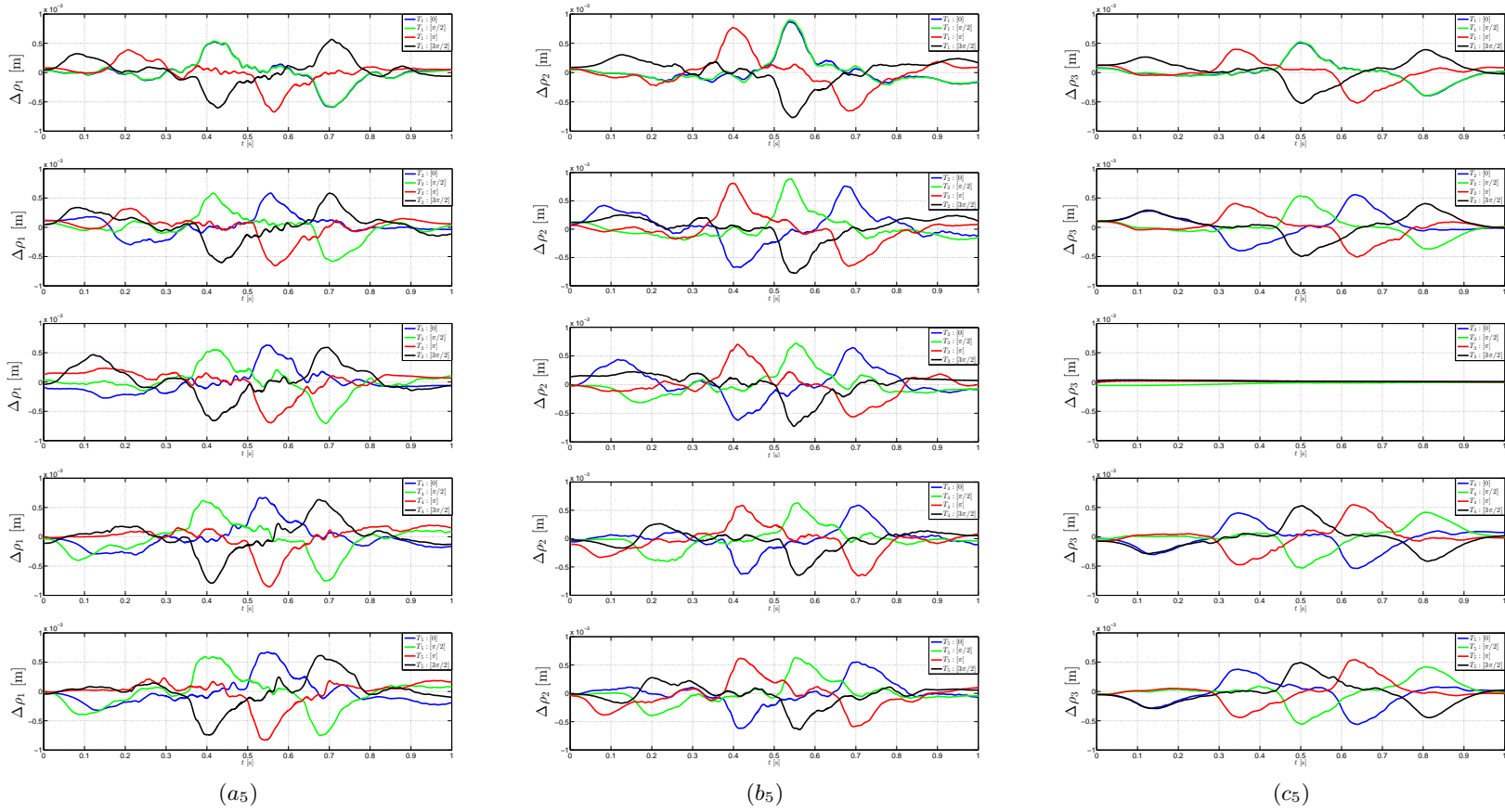


Figure V.11 – Joint error values, $\Delta\rho_j$ from experimental data for different starting position [$T_i : [0]$ (in blue), $T_i : [\pi/2]$ (in green), $T_i : [\pi]$ (in red), $T_i : [3\pi/2]$ (in black)] of Orthoglide 5-axis along trajectories $\phi_1(t)$, $\phi_2(t)$, $\phi_3(t)$, $\phi_4(t)$, $\phi_5(t)$ for ρ_1 (a) ρ_2 (b) ρ_3 (c), where $i = 1, 2, 3, 4, 5$ which represents different trajectories and $j = 1, 2, 3$.

Table V.3 – The Minimum, Maximum and mean absolute values of joint errors along the trajectories with different starting position

Errors (μm) \rightarrow Trajectories \downarrow	Starting Point	$\Delta\rho_1$			$\Delta\rho_2$			$\Delta\rho_3$		
		Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
T_1 : [-80, -80, -140]	0	-591.5	529.3	129.7	-196.2	866.4	136.2	-396.3	506.8	112.9
	$\pi/2$	-583.8	538.5	126.2	-204.9	897.8	143.5	-392.3	524.6	113.1
	π	-673.1	387.1	121.2	-655.1	765.4	185.3	-518.4	397.4	126.2
	$3\pi/2$	-603.1	565.9	158.5	-769.0	304.7	172.5	-524.1	390.5	151.4
T_2 : [-80, -80, -80]	0	-297.3	586.0	125.5	-676.7	761.0	217.4	-400.5	557.4	159.4
	$\frac{\pi}{2}$	-583.7	585.3	140.0	-195.2	888.9	148.0	-372.2	536.4	114.1
	π	-606.0	582.5	175.0	-778.1	251.5	168.5	-495.7	404.7	154.9
	$3\pi/2$	-312.7	626.6	130.6	-665.2	777.7	220.2	-524.1	529.3	151.4
T_3 : [00, 00, 00]	0	-272.9	631.2	141.7	-622.0	644.2	199.1	002.4	033.7	016.5
	$\pi/2$	-707.1	551.6	145.5	-311.6	719.3	151.7	-059.0	-00.7	024.9
	π	-697.3	236.6	147.8	-563.0	703.3	167.6	-008.9	021.2	010.3
	$3\pi/2$	-662.1	592.1	197.5	-727.9	224.3	149.5	003.5	033.7	016.9
T_4 : [80, 80, 80]	0	-309.5	669.3	158.4	-626.8	589.3	136.7	-540.7	406.3	157.3
	$\pi/2$	-755.1	618.9	201.9	-401.2	631.8	128.5	-532.9	417.6	121.9
	π	-857.3	196.3	134.6	-660.4	586.6	165.0	-477.6	544.9	135.3
	$3\pi/2$	-793.9	633.2	176.0	-648.1	261.5	123.5	-419.2	529.9	150.7
T_5 : [80, 80, 140]	0	-322.9	674.0	168.7	-619.1	552.9	140.1	-560.2	379.3	158.0
	$\pi/2$	-751.6	595.0	202.9	-392.7	631.7	129.4	-559.0	419.0	122.5
	π	-829.4	230.0	136.5	-587.6	619.0	179.2	-445.2	540.7	126.4
	$3\pi/2$	-744.0	614.5	165.2	-640.1	277.7	124.7	-446.8	492.6	154.9

maximum error in ρ_2 is more [888.9 μm] for the trajectory $T_2 : [\pi/2]$, whereas less value [251.5 μm] for the trajectory $T_2 : [\pi]$. The mean absolute error in ρ_2 is the maximum [220.2 μm] for the trajectory $T_2 : [3\pi/2]$ and minimum [148 μm] for the trajectory $T_2 : [\pi/2]$. The maximum error in ρ_3 is more [557.4 μm] for the trajectory $T_2 : [0]$, whereas less [404.7 μm] for the trajectory $T_2 : [\pi]$. The mean absolute error in ρ_3 is the maximum [159.4 μm] for the trajectory $T_2 : [0]$ and minimum [114.1 μm] for the trajectory $T_2 : [\pi/2]$.

Trajectory 3: T_3

The maximum error in ρ_1 is more [631.2 μm] for the starting position $T_3 : [0]$, whereas less [236.6 μm] for the trajectory $T_3 : [\pi]$. The mean absolute error in ρ_1 is the maximum [197.5 μm] for the trajectory $T_3 : [3\pi/2]$ and minimum [141.7 μm] for the trajectory $T_3 : [0]$. The maximum error in ρ_2 is more [719.3 μm] for the trajectory $T_3 : [\pi/2]$, whereas less value [224.3 μm] for the trajectory $T_3 : [3\pi/2]$. The mean absolute error in ρ_2 is the maximum [199.1 μm] for the trajectory $T_3 : [0]$ and minimum [151.7 μm] for the trajectory $T_3 : [\pi/2]$. The maximum error in ρ_3 is more [33.7 μm] for the trajectory $T_3 : [0]$, whereas less [-0.7 μm] for the trajectory $T_3 : [\pi/2]$. The mean absolute error in ρ_3 is the maximum [24.9 μm] for the trajectory $T_3 : [\pi/2]$ and minimum [10.3 μm] for the trajectory $T_3 : [\pi]$.

Trajectory 4: T_4

The maximum error in ρ_1 is more [699.3 μm] for the starting position $T_4 : [0]$, whereas less [196.3 μm] for the trajectory $T_4 : [\pi]$. The mean absolute error in ρ_1 is the maximum [201.9 μm] for the trajectory $T_4 : [\pi/2]$ and minimum [134.6 μm] for the trajectory $T_4 : [\pi]$. The maximum error in ρ_2 is more [631.8 μm] for the trajectory $T_4 : [\pi/2]$, whereas less value [261.5 μm] for the trajectory $T_4 : [3\pi/2]$. The mean absolute error in ρ_2 is the maximum [165 μm] for the trajectory $T_4 : [\pi]$ and minimum [123.5 μm] for the trajectory $T_4 : [3\pi/2]$. The maximum error in ρ_3 is more [544.9 μm] for the trajectory $T_4 : [\pi]$, whereas less [406.3 μm] for the trajectory $T_4 : [0]$. The mean absolute error in ρ_3 is the maximum [157.3 μm] for the trajectory $T_4 : [0]$ and minimum [121.9 μm] for the trajectory $T_4 : [\pi/2]$.

Trajectory 5: T_5

The maximum error in ρ_1 is more [674 μm] for the starting position $T_5 : [0]$, whereas less [230 μm] for the trajectory $T_5 : [\pi]$. The mean absolute error in ρ_1 is the maximum [202.9 μm] for the trajectory $T_5 : [\pi/2]$ and minimum [136.5 μm] for the trajectory $T_5 : [\pi]$. The maximum error in ρ_2 is more [631.7 μm] for the trajectory $T_5 : [\pi/2]$, whereas less value [277.7 μm] for the trajectory $T_5 : [3\pi/2]$. The mean absolute error in ρ_2 is the maximum [179.2 μm] for the trajectory $T_5 : [\pi]$ and minimum [124.7 μm] for the trajectory $T_5 : [3\pi/2]$. The maximum error in ρ_3 is more [540.7 μm] for the trajectory $T_5 : [\pi]$, whereas less [379.3 μm] for the trajectory $T_5 : [0]$. The mean absolute error in ρ_3 is the maximum [158 μm] for the trajectory $T_5 : [0]$ and minimum [122.5 μm] for the trajectory $T_5 : [\pi/2]$.

It is difficult to choose the best suitable location for a trajectory based on the joint errors because we should evaluate their images in the workspace. For T_3 , the joint error is minimum, as there is no motion along ρ_3 .

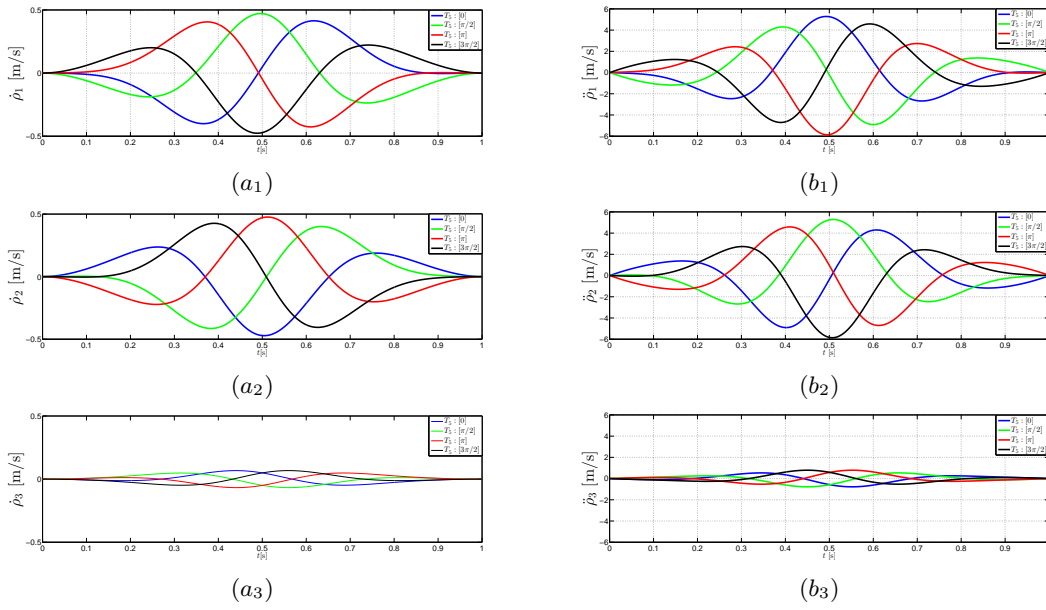


Figure V.12 – Cartesian velocities, $\dot{\rho}_i$ (a_i) and Cartesian accelerations, $\ddot{\rho}_i$ (b_i) along trajectory $\phi_5(t)$ for different starting positions [$T_5 : [0]$ (in blue), $T_5 : [\pi/2]$ (in green), $T_5 : [\pi]$ (in red), $T_5 : [3\pi/2]$ (in black)].

V.7.2 Projection of Joint errors in the Cartesian space

The Cartesian velocities and accelerations along trajectory $\phi_5(t)$ for the different starting positions [$T_5 : [0]$ (in blue), $T_5 : [\pi/2]$ (in green), $T_5 : [\pi]$ (in red), $T_5 : [3\pi/2]$ (in black)] is shown in Fig. V.12. Newton method is used to compute the image of joint errors in the workspace of the manipulator. By differentiating Eq. (V.3), i.e, inverse kinematic equations and substituting the values of proposed error model $\Delta\rho_i^m$, i.e $e_1 = \Delta\rho_1^m$, $e_2 = \Delta\rho_2^m$ and $e_3 = \Delta\rho_3^m$ in Eq. (V.15), we will get the image of joint errors Δx , Δy and Δz in the workspace of the manipulator. The projected joint errors in the workspace Δx , Δy and Δz along the the trajectories $\phi_5(t)$ for the different starting positions is shown in Fig. V.13. A comparison of projection of joint errors in the Cartesian space for trajectories $\phi_i(t)$ $i \in [1..5]$ with different starting positions of the manipulator is shown in Fig. V.14.

The main aim of the experiment is to study the influence of starting position of the trajectory on the accuracy of the manipulator. Experimental data from the sensor proves that there is a significant change in the joint errors during the execution of these trajectories.. The proposed error model, which is based on the joint velocities and acceleration enables us to project the joints error in the workspace more accurately unlike other model based on Jacobian or condition number. By analyzing these projected errors in the workspace, We have found out that the trajectory closer to the singularity surface, T_5 with starting point $T_5 : [\pi/2]$ admitted the maximum error and is approximately five times than the trajectory closest to the isotropic posture, T_3 . The acceleration parameter in the proposed model helps to compute the difference in the accuracy more precisely unlike other technique based on Jacobian.

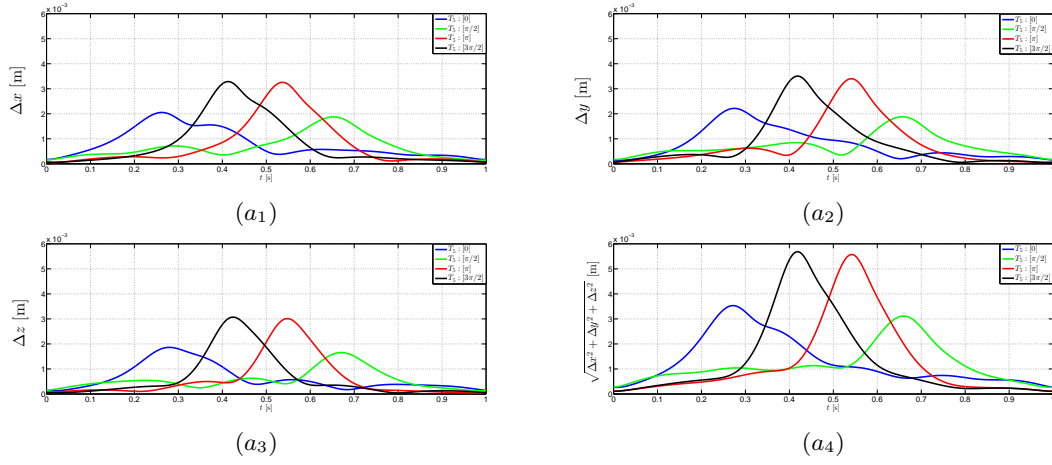


Figure V.13 – Cartesian errors in \mathbf{x} (a_1), \mathbf{y} (a_2), \mathbf{z} (a_3) and norm (a_4) along the trajectories $\phi_5(t)$ for different starting positions [$T_1 : [0]$ (in blue), $T_2 : [\pi/2]$ (in green), $T_3 : [\pi]$ (in red), $T_4 : [3\pi/2]$ (in black)], Image of the joints errors in the workspace of the manipulator.

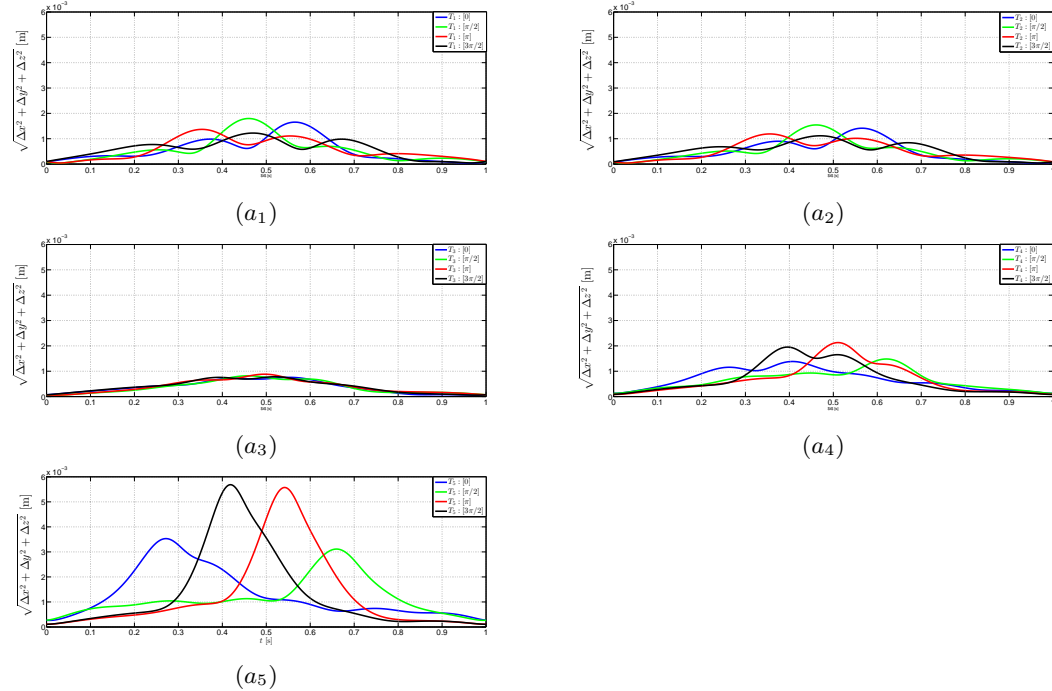


Figure V.14 – Cartesian errors, norm, along trajectories $\phi_1(t)$ (a_1), $\phi_2(t)$ (a_2), $\phi_3(t)$ (a_3), $\phi_4(t)$ (a_4) and $\phi_5(t)$ (a_5) for different starting positions [$T_1 : [0]$ (in blue), $T_2 : [\pi/2]$ (in green), $T_3 : [\pi]$ (in red), $T_4 : [3\pi/2]$ (in black)], Image of the joints errors in the workspace of the manipulator.

V.8 Conclusions

This chapter dealt with the error modeling of the Orthoglide 5-axis thanks to experimental data associated with the trajectory planning parameters. The accuracy of the path in the Cartesian workspace was studied for five circular trajectories with the same velocity and acceleration profiles. The proposed joint error model which is based on the dynamic properties i.e joint velocities and acceleration of the Orthoglide helps in estimating the error in the Cartesian workspace. We have found out that the trajectory closer to the singularity surface, T_5 admitted the maximum error and is approximately twice than the trajectory closest to the isotropic posture, T_3 . Further, from the experimental results it is observed that there is a significant change in the accuracy by changing the starting position of the manipulator. Finally, from the analysis, it can be concluded that by selecting the appropriate location and starting point, the accuracy of the manipulator can be enhanced significantly. Among the five trajectories, which are defined for the analysis, trajectory T_5 has approximately five times pose error than the trajectory T_3 , which is closest to the isotropic posture. For a given trajectory, we can search the optimal starting point and location for the trajectory so that error in the pose of end effector can be minimized.

With this error model, it is possible to make the optimal path placement or to modify the speed/acceleration settings in the trajectory generation to respect a fixed machining accuracy. Experiments are ongoing to refine the accuracy model by changing the maximum velocity and acceleration and relate these to physical phenomena such as frictions in the joints. In parallel to this work, the dynamic parameters of the robot identification work is in progress to improve the dynamic model. The encoders are also being changed because there is noise in the velocity estimate that comes from this encoder emulation in drives.

VI

Conclusions and Future Work

VI.1 Summary

This thesis focuses on the different aspects which are associated with efficient planning of desired tasks for parallel robots. These different aspects are mainly categorized in four parts, namely: workspace and joint space analysis, uniqueness domains, trajectory planning and accuracy analysis. The workspace and joint space analysis differentiate the regions with different number of inverse kinematic solutions and direct kinematic solutions using a cylindrical algebraic decomposition algorithm, respectively. The influence of design parameters and joint limits on the workspace boundaries for the parallel robots are reported. Gröbner based elimination methods are used to compute the parallel and serial singularities of the manipulator under study. The descriptive analysis of a family of delta like robots is presented by using algebraic tools to induce the an estimation about the complexity in representing the singularities in the workspace and the joint space. The generalized notions of aspects and uniqueness domains are defined for the parallel robot with several operation modes. The characteristic surfaces are also computed to define the uniqueness domains in the workspace. An algebraic method is proposed to check the feasibility of any given trajectory in the workspace to address the well known problem which arises when there exists a singular configuration between the two poses of the end-effectors while discretizing the path with a classical approach. A Framework for the control loop of a parallel robot with several actuation modes is presented , which uses only the inverse geometric model. The accuracy analysis focuses on the estimation of errors in the pose of the end effector due to the joint's errors produced by the PID control loop. The proposed error model, which is based on the static and dynamic properties of the Orthoglide, helps in estimating the error in the Cartesian workspace.

VI.2 Main Contributions

1- Workspace and joint space analysis of the 3-RPS parallel robot

This work reports the variations in the workspace, singularities and joint space with respect to the design parameter ' k ' of the 3-RPS parallel manipulator. The cylindrical algebraic decomposition method and Gröbner based computations are used to model the workspace and the joint space with the parallel singularities in 3T & 2R1T projection spaces, where

the orientation of the mobile platform is represented by using quaternions. An algorithm is presented to separate the singularity surfaces for the positive and negative values of quaternion q_1 & q_4 for the corresponding operation modes. Depending on the design parameter 'k', three different configurations of the 3-RPS parallel manipulator are analyzed.

2- The uniqueness domains for parallel robot with several operation modes and assembly modes

In this contribution, we have studied the joint space and the workspace of the 3-RPS parallel robot. The joint space analysis is conceded to define the regions where the direct kinematic problem admits a constant number of solutions and the location of the cusp points. With the presented two operation modes, the aspects were set to define the maximal singularity free region of the workspace. Each operation mode is divided into two aspects which further show that non-singular assembly mode changing trajectories are possible. Basic regions can be defined for each operation mode by computing the characteristic surfaces. With the knowledge of basic regions defined in the workspace, the uniqueness domains can be illustrated in a slice of the workspace. This method can be used for the optimization of workspace's size or the trajectory planning.

3- Non-singular assembly mode changing trajectories in the workspace for the 3-RPS parallel robot

This part presents a study of the joint space and workspace of the 3-RPS parallel robot and shows the existence of non-singular assembly mode changing trajectories. First, it has been shown that each of the two operation modes is divided into two aspects, which is a necessary condition for knowing non-singular assembly mode changing trajectories. Moreover, it turns out that this mechanism has a maximum of 16 real solutions to the direct kinematic problem, eight for each operation mode. By computing the characteristic surfaces, we have shown that we can describe the basic regions for each operation mode. We construct a path going through several basic regions which are images of the same basic component with 8 solutions for the DKP. The analysis of the determinant of Jacobian shows that a non-singular assembly mode change exists for each motion type.

4- An Algebraic Method to Check the Singularity-Free Paths for Parallel Robots

This work reports the use of algebraic methods to check the feasibility of given trajectories in the workspace. The general method uses the projection of the solutions of the polynomial equations associated with the trajectories in the joint space using a Gröbner based elimination strategy. There is a significant change in the shape of workspace and in the singularity surfaces due to the joint constraints. The proposed method enables us to project the joint constraints in the workspace of the manipulator, which further helps to analyse the projection of the singularities in the workspace with the joint constraints. Such computations ensure the existence of a singular configuration between two poses of the end-effector unlike other numerical or discretization methods. This work highlights that the singularity

analysis should be done in the cross product of the workspace and joint space for parallel robots with several assembly and working modes. The single analysis of the singularities in a projection space can introduce spurious singularities. In fact, the algebraic tools does not allow to distinguish the parallel singularities according to the working mode.

5- Workspace and Singularity analysis of a Delta like family robot

This contribution presents the descriptive analysis of a family of delta-like parallel robots by using algebraic tools to induce an estimation about the complexity in representing the singularities in the workspace and the joint space. A comparative study on the workspace of different delta-like robots gives the idea about the shape of the workspace, which further plays an important role in the selection of the manipulator for the specific task or for the trajectory planning. The main characteristics associated with the singularities are tabulated in Table ??, which also gives some information about the complexity of the singularities, which is an essential factor for the singularity-free path plannings. From these data, it can be observed that the singularities associated with the Hybridglide are complicated, whereas the structure of those associated with the Triaglide is rather simple.

6- Influence of the trajectory planning on the accuracy of the Orthoglide 5-axis

This contribution dealt with the error modeling of the Orthoglide 5-axis thanks to experimental data associated with the trajectory planning parameters. This work reports the influence of static and dynamic parameters on the computation of the error in the pose associated with the trajectory planning made and analyzed with the Orthoglide 5-axis. The accuracy of the path in the Cartesian workspace was studied for five circular trajectories with the same velocity and acceleration profiles. The proposed joint error model which is based on the dynamic properties i.e joint velocities and acceleration of the Orthoglide helps in estimating the error in the Cartesian workspace. We have found out that the trajectory closer to the singularity surface admitted the maximum error and is approximately twice than the trajectory closest to the isotropic posture. Further, from the experimental results it is observed that there is a significant change in the accuracy by changing the starting position of the manipulator. Finally, from the analysis, it can be concluded that by selecting the appropriate location and starting point, the accuracy of the manipulator can be enhanced significantly. Among the five trajectories, which are defined for the analysis, trajectory near singularity surface has approximately five times pose error than the trajectory, which is closest to the isotropic posture. For a given trajectory, we can search the optimal starting point and location for the trajectory so that error in the pose of end effector can be minimized.

With this error model, it is possible to make the optimal path placement or to modify the speed/acceleration settings in the trajectory generation to respect a fixed machining accuracy. Experiments are ongoing to refine the accuracy model by changing the maximum velocity and acceleration and relate these to physical phenomena such as frictions in the joints. In parallel to this work, the dynamic parameters of the robot identification work is

in progress to improve the dynamic model. The encoders are also being changed because there is noise in the velocity estimate that comes from this encoder emulation in drives.

7- A Framework for the Control Loop of a Parallel Robot with Several Actuation Modes

With the NaVARo robot, a variable actuation robot, it is possible to introduce all the problems related to the operating principle as the loss of connection between the encoder and the frames of the robot. A start procedure allows to identify the current assembly mode of the robot by using magnetic sensors and absolute encoders for low accuracy. The movements sensed by the magnetic sensors at low speed and always in the same direction can guarantee the reproducibility of the procedure. A framework is presented to realize the trajectory that does not use the direct geometric model, which uses only the inverse geometric model. The redundant sensors ensure that the current method of assembly for the command is the one used by the robot. This allows to perform assembly mode change paths and confirms that the movement has been achieved.

VI.3 Future Work

1- Workspace and singularity analysis of other higher degrees of freedom and complex Parallel robots

The workspace and joint space analysis differentiate the regions with different number of inverse kinematic solutions and direct kinematic solutions using cylindrical algebraic decomposition algorithm, respectively. The influence of design parameters and joint limits on the workspace boundaries and singularities for 3RPS and delta like family parallel robots are presented in the Chapter 2 and 3. Gröbner based elimination methods are used to compute the parallel and serial singularities of the parallel robots. Workspace and singularity analysis of other similar or more complex mechanisms can be made using these methodologies and algebraic tools.

2- Algorithm to find the connected regions in the projection spaces

The singularities divide the workspace into aspects and the characteristic surfaces induce a partition of each aspect into the basic regions. For the parallel robots with several inverse and direct kinematic solutions, the aspects are defined as the maximal singularity-free connected sets in the workspace or the cross-product of the joint space by the workspace. The connectivity of the cells in three dimensional space using the cylindrical algebraic decomposition algorithm can help to find these connected regions inside any projection space. This information can help to make non singular assembly mode changing trajectories or singularity-free path planning in the workspace.

3- Algorithm to generate optimized singularity-free trajectory to reach from one given pose to another within the workspace

One of the important steps in the trajectory planning is to check the singularity-free paths in the workspace for the parallel robots. The main problem arises when there exists a singular configuration between two poses of the end-effectors while discretizing the path with a classical approach. In Chapter 4, a technique based on some algebraic methods to check the feasibility of any given trajectories in the workspace is presented, it allows to write the Jacobian of the manipulator as a function of time and to check whether its determinant vanishes between two poses. Also, when the trajectory meets a singularity, its location can also be computed. In the future, the same methodology with the knowledge of the connectivity of the cells in the three dimensional space can be extended to generate optimized singularity-free paths between the two poses, which is shown in Fig. VI.1(b). In Fig. VI.1(a), a singularity surface of 3RPS parallel robot in $[pz, q2, q3]$ projection space is shown.

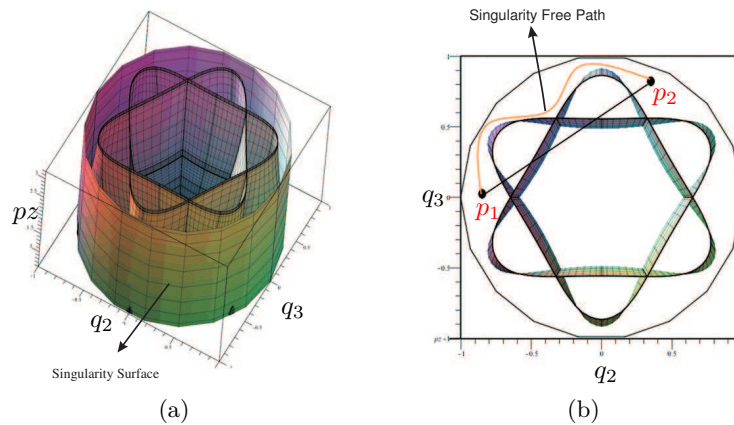


Figure VI.1 – Parallel singularity surface of 3RPS parallel robot (a) The methodology can be extended to find the optimized path between point p_1 and p_2 (b) which is shown in red color.

4- Influence of stiffness on the accuracy of the manipulator

One of the highly addressed problems associated with the end-effector pose error is the manipulator stiffness, which defines the positioning error due to the external loading while executing a specific task by the manipulator. In Chapter 5, the proposed error model which is based on the dynamic properties (joint velocities and acceleration) of the Orthoglide helps in estimating the error in the Cartesian workspace. This error model can be extended by including the stiffness parameter for estimating the error in the workspace.

5- Projection of joints error in the workspace

There are some limitations to the methodology which is described in the Chapter 5, while projecting the joints error in the workspace for parallel robots which involves the complex equations in the analysis. One such example is shown in Fig. VI.1. The work can be extended

for such robots by including the orientation motions to compute the projection of joints error in the workspace.

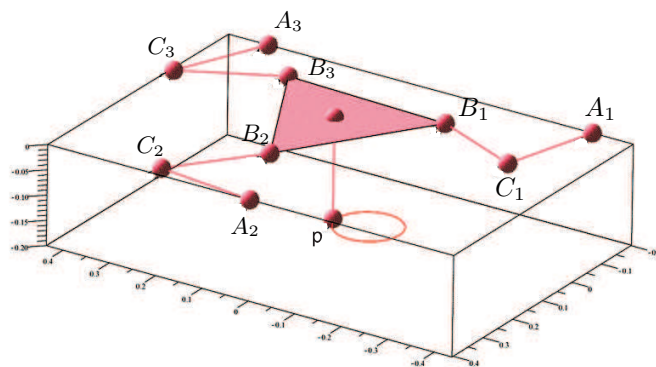


Figure VI.2 – The virtual model of the 3RRR robot

Personal Publications

Peer-Reviewed Publications

1. Ranjan Jha, Damien Chablat, Fabrice Rouillier, Guillaume Moroz. ***Influence of the trajectory planning on the accuracy of the orthoglide 5-axis***. ASME International Design Engineering Technical Conference and the Computer and Information in Engineering Conference (IDETC/CIE), Aug 2016, Charlotte, NC, United States. Proceedings of the ASME 2016 International Design Engineering Technical Conference and the Computer and Information in Engineering Conference (IDETC/CIE), 2016. <hal-01309190>
2. Damien Chablat, Ranjan Jha, Stephane Caro. A Framework for the Control of a Parallel Manipulator with Several Actuation Modes. 14th International Conference on Industrial Informatics , Jul 2016, Poitiers, France. IEEE, 2016. <hal-01313376>
3. Ranjan Jha, Damien Chablat, Fabrice Rouillier, Guillaume Moroz. ***Workspace and Singularity analysis of a Delta like family robot***. 4th IFTOMM International Symposium on Robotics and Mechatronics, Jun 2015, Poitiers, France. 2015. <hal-01142465>
4. Ranjan Jha, Damien Chablat, Fabrice Rouillier, Guillaume Moroz. ***An algebraic method to check the singularity-free paths for parallel robots***. International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Aug 2015, Boston, United States. 2015. <hal-01142989>
5. Damien Chablat, Ranjan Jha, Fabrice Rouillier, Guillaume Moroz. ***Workspace and joint space analysis of the 3-RPS parallel robot***. ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Aug 2014, Buffalo, United States. Volume 5A, pp.1-10, 2014. <hal-01006614>
6. Damien Chablat, Ranjan Jha, Fabrice Rouillier, Guillaume Moroz. ***Non-singular assembly mode changing trajectories in the workspace for the 3-RPS parallel robot***. 14th International Symposium on Advances in Robot Kinematics, Jun 2014, Ljubljana, Slovenia. pp.149 - 159, 2014. <hal-00956325>
7. Ranjan Jha, ***Singularity-Free Trajectory and Workspace Analysis for Parallel Robots Using Algebraic Tool***. 16th Annual Day of ED STIM. [won 2nd prize for presentation].

Posters

1. Jha, Ranjan, Damien Chablat, Fabrice Rouillier, and Guillaume Moroz. "*A partition of the workspace and jointspace of parallel robots using algebraic tools.*" Applications of Parallel and Cable-driven Robots, INNOROBO, 2014.



Mathematical Expressions

A.1 Manipulator Creation

SIROPA library Provides modeling, analyzing and plotting functions for different manipulators.

ORTHOGLIDE

Record[

Equations = $[\rho_1^2 - 2\rho_1x + x^2 + y^2 + z^2 - 4, \rho_2^2 - 2\rho_2y + x^2 + y^2 + z^2 - 4, \rho_3^2 - 2\rho_3z + x^2 + y^2 + z^2 - 4]$, **Constraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **PoseVariables** = $[x, y, z]$, **ArticularVariables** = $[\rho_1, \rho_2, \rho_3]$, **PassiveVariables** = $[]$, **GeometricParameters** = $[]$, **GenericEquations** = $[(x - \rho_1)^2 + y^2 + z^2 - L^2, x^2 + (y - \rho_2)^2 + z^2 - L^2, x^2 + y^2 + (z - \rho_3)^2 - L^2]$, **GenericConstraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **Precision** = 4, **PoseValues** = $[]$, **ArticularValues** = $[]$, **PassiveValues** = $[]$, **GeometricValues** = $[L = 2]$, **DefaultPlotRanges** = $[]$, **Points** = $[A1 = [2L, 0, 0], A2 = [0, 2L, 0], A3 = [0, 0, 2L], B1 = [\rho_1, 0, 0], B2 = [0, \rho_2, 0], B3 = [0, 0, \rho_3], P = [x, y, z]]$, **Loops** = $[]$, **Chains** = $[[B1, P], [B2, P], [B3, P]]$, **Actuators** = $[[A1, B1], [A2, B2], [A3, B3]]$, **Model** = "Orthoglide"
]

TRIAGLIDE

Record[

Equations = $[\rho_1^2 - 2\rho_1y + x^2 + y^2 + z^2 - 2x - 3, \rho_2^2 - 2\rho_2y + x^2 + y^2 + z^2 + 2x - 3, \rho_3^2 - 2\rho_3y + x^2 + y^2 + z^2 - 4]$, **Constraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **PoseVariables** = $[x, y, z]$, **ArticularVariables** = $[\rho_1, \rho_2, \rho_3]$, **PassiveVariables** = $[]$, **GeometricParameters** = $[]$, **GenericEquations** = $[(x - 1)^2 + (y - \rho_1)^2 + z^2 - L^2, (x + 1)^2 + (y - \rho_2)^2 + z^2 - L^2, x^2 + (y - \rho_3)^2 + z^2 - L^2]$, **GenericConstraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **Precision** = 4, **PoseValues** = $[]$, **ArticularValues** = $[]$, **PassiveValues** = $[]$, **GeometricValues** = $[L = 2]$, **DefaultPlotRanges** = $[]$, **Points** = $[A1 = [1, 0, 0], A2 = [-1, 0, 0], A3 = [0, 4, 0], B1 = [1, \rho_1, 0], B2 = [-1, \rho_2, 0], B3 = [0, \rho_3, 0], P = [x, y, z]]$, **Loops** = $[]$, **Chains** = $[[B1, P], [B2, P], [B3, P]]$, **Actuators** = $[[A1, B1], [A2, B2], [A3, B3]]$, **Model** = "Triaglide"
]

HYBRIDGLIDE

Record[

Equations = $[\rho_1^2 - 2\rho_1y + x^2 + y^2 + z^2 - 2x - 3, \rho_2^2 - 2\rho_2y + x^2 + y^2 + z^2 + 2x - 3, \rho_3^2 - 2\rho_3z + x^2 + y^2 + z^2 - 4]$, **Constraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **PoseVariables** = $[x, y, z]$, **ArticularVariables** = $[\rho_1, \rho_2, \rho_3]$, **PassiveVariables** = $[\]$, **GeometricParameters** = $[\]$, **GenericEquations** = $[(x-1)^2 + (y-\rho_1)^2 + z^2 - L^2, (x+1)^2 + (y-\rho_2)^2 + z^2 - L^2, x^2 + y^2 + (z-\rho_3)^2 - L^2]$, **GenericConstraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **Precision** = 4, **PoseValues** = $[\]$, **ArticularValues** = $[\]$, **PassiveValues** = $[\]$, **GeometricValues** = $[L = 2]$, **DefaultPlotRanges** = $[\]$, **Points** = $[A1 = [1, 0, 0], A2 = [-1, 0, 0], A3 = [0, 0, 0], B1 = [1, \rho_1, 0], B2 = [-1, \rho_2, 0], B3 = [0, 0, \rho_3], P = [x, y, z]]$, **Loops** = $[\]$, **Chains** = $[[B1, P], [B2, P], [B3, P]]$, **Actuators** = $[[A1, B1], [A2, B2], [A3, B3]]$, **Model** = "Hybridglide"]

UraneSX

Record[

Equations = $[\rho_1^2 - 2\rho_1z + x^2 + y^2 + z^2 - 2x - 3, x^2 + x - 120071/40000 + y^2 - (173/100)y + \rho_2^2 - 2z\rho_2 + z^2, x^2 + x - 120071/40000 + y^2 + (173/100)y + \rho_3^2 - 2z\rho_3 + z^2]$, **Constraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **PoseVariables** = $[x, y, z]$, **ArticularVariables** = $[\rho_1, \rho_2, \rho_3]$, **PassiveVariables** = $[\]$, **GeometricParameters** = $[\]$, **GenericEquations** = $[(x-1)^2 + y^2 + (z-\rho_1)^2 - L^2, (x+1/2)^2 + (y-173/200)^2 + (z-\rho_2)^2 - L^2, (x+1/2)^2 + (y+173/200)^2 + (z-\rho_3)^2 - L^2]$, **GenericConstraints** = $[0 < \rho_1, 0 < \rho_2, 0 < \rho_3, 0 < 4 - \rho_1, 0 < 4 - \rho_2, 0 < 4 - \rho_3]$, **Precision** = 4, **PoseValues** = $[\]$, **ArticularValues** = $[\]$, **PassiveValues** = $[\]$, **GeometricValues** = $[L = 2]$, **DefaultPlotRanges** = $[\]$, **Points** = $[A1 = [1, 0, 0], A2 = [-1/2, 173/200, 0], A3 = [-1/2, -173/200, 0], B1 = [1, 0, \rho_1], B2 = [-1/2, 173/200, \rho_2], B3 = [-1/2, -173/200, \rho_3], P = [x, y, z]]$, **Loops** = $[\]$, **Chains** = $[[B1, P], [B2, P], [B3, P]]$, **Actuators** = $[[A1, B1], [A2, B2], [A3, B3]]$, **Model** = "UraneSX"]

A.2 Parallel Singularities

Projection in workspace

$$s2^{Ort} = [x^10y^4z^4 + 3x^8y^6z^4 + 3x^8y^4z^6 + 3x^6y^8z^4 + 6x^6y^6z^6 + 3x^6y^4z^8 + x^4y^10z^4 + 3x^4y^8z^6 + 3x^4y^6z^8 + x^4y^4z^10 - 28x^8y^4z^4 - 56x^6y^6z^4 - 56x^6y^4z^6 - 28x^4y^8z^4 - 56x^4y^6z^6 - 28x^4y^4z^8 - 8x^10y^2z^2 + 32x^8y^4z^2 + 32x^8y^2z^4 + 80x^6y^6z^2 + 400x^6y^4z^4 + 80x^6y^2z^6 + 32x^4y^8z^2 + 400x^4y^6z^4 + 400x^4y^4z^6 + 32x^4y^2z^8 - 8x^2y^10z^2 + 32x^2y^8z^4 + 80x^2y^6z^6 + 32x^2y^4z^8 - 8x^2y^2z^10 - 64x^8y^2z^2 - 704x^6y^4z^2 - 704x^6y^2z^4 - 704x^4y^6z^2 - 2240x^4y^4z^4 - 704x^4y^2z^6 - 64x^2y^8z^2 - 704x^2y^6z^4 - 704x^2y^4z^6 - 64x^2y^2z^8 + 16x^10 + 80x^8y^2 + 80x^8z^2 + 160x^6y^4 + 1856x^6y^2z^2 + 160x^6z^4 + 160x^4y^6 + 4576x^4y^4z^2 + 4576x^4y^2z^4 + 160x^4z^6 + 80x^2y^8 + 1856x^2y^6z^2 + 4576x^2y^4z^4 + 1856x^2y^2z^6 + 80x^2z^8 + 16y^10 + 80y^8z^2 + 160y^6z^4 + 160y^4z^6 + 80y^2z^8 + 16z^10 - 320x^8 - 1280x^6y^2 - 1280x^6z^2 - 1920x^4y^4 - 11008x^4y^2z^2 - 1920x^4z^4 - 1280x^2y^6 - 11008x^2y^4z^2 - 11008x^2y^2z^4 - 1280x^2z^6 -$$

$$320y^8 - 1280y^6z^2 - 1920y^4z^4 - 1280y^2z^6 - 320z^8 + 2560x^6 + 7680x^4y^2 + 7680x^4z^2 + 7680x^2y^4 + 25600x^2y^2z^2 + 7680x^2z^4 + 2560y^6 + 7680y^4z^2 + 7680y^2z^4 + 2560z^6 - 10240x^4 - 20480x^2y^2 - 20480x^2z^2 - 10240y^4 - 20480y^2z^2 - 10240z^4 + 20480x^2 + 20480y^2 + 20480z^2 - 16384]$$

$$s2_w^{Tri} = [z(z^2 - 4)]$$

$$s2_w^{Hyb} = [x^16z^4 + 4x^14y^2z^4 + 4x^14z^6 + 6x^12y^4z^4 + 12x^12y^2z^6 + 6x^12z^8 + 4x^10y^6z^4 + 12x^10y^4z^6 + 12x^10y^2z^8 + 4x^10z^10 + x^8y^8z^4 + 4x^8y^6z^6 + 6x^8y^4z^8 + 4x^8y^2z^10 + x^8z^12 - 28x^14z^4 - 80x^12y^2z^4 - 84x^12z^6 - 72x^10y^4z^4 - 156x^10y^2z^6 - 92x^10z^8 - 16x^8y^6z^4 - 60x^8y^4z^6 - 88x^8y^2z^8 - 44x^8z^10 + 4x^6y^8z^4 + 12x^6y^6z^6 + 4x^6y^4z^8 - 12x^6y^2z^10 - 8x^6z^12 + 32x^14z^2 + 64x^12y^2z^2 + 422x^12z^4 + 784x^10y^2z^4 + 828x^10z^6 - 64x^8y^6z^2 + 300x^8y^4z^4 + 940x^8y^2z^6 + 630x^8z^8 - 32x^6y^8z^2 - 64x^6y^6z^4 + 164x^6y^4z^6 + 372x^6y^2z^8 + 192x^6z^10 - 2x^4y^8z^4 + 52x^4y^6z^6 + 94x^4y^4z^8 + 40x^4y^2z^10 + 16x^4z^12 - 704x^12z^2 - 1408x^10y^2z^2 - 3556x^10z^4 - 768x^8y^4z^2 - 5016x^8y^2z^4 - 4428x^8z^6 - 128x^6y^6z^2 - 1936x^6y^4z^4 - 3892x^6y^2z^6 - 2096x^6z^8 - 64x^4y^8z^2 - 488x^4y^6z^4 - 1284x^4y^4z^6 - 968x^4y^2z^8 - 320x^4z^10 - 12x^2y^8z^4 - 28x^2y^6z^6 + 8x^2y^4z^8 + 32x^2y^2z^10 + 256x^12 + 1024x^10y^2 + 6688x^10z^2 + 1536x^8y^4 + 12480x^8y^2z^2 + 17041x^8z^4 + 1024x^6y^6 + 8064x^6y^4z^2 + 19724x^6y^2z^4 + 12608x^6z^6 + 256x^4y^8 + 2368x^4y^6z^2 + 7406x^4y^4z^4 + 8120x^4y^2z^6 + 2912x^4z^8 + 96x^2y^8z^2 + 380x^2y^6z^4 + 272x^2y^4z^6 - 160x^2y^2z^8 + 9y^8z^4 + 24y^6z^6 + 16y^4z^8 - 4096x^10 - 12288x^8y^2 - 33152x^8z^2 - 12288x^6y^4 - 49024x^6y^2z^2 - 43976x^6z^4 - 4096x^4y^6 - 21120x^4y^4z^2 - 33496x^4y^2z^4 - 15168x^4z^6 - 1152x^2y^6z^2 - 2392x^2y^4z^4 - 544x^2y^2z^6 - 72y^6z^4 - 96y^4z^6 + 24576x^8 + 49152x^6y^2 + 83456x^6z^2 + 24576x^4y^4 + 71680x^4y^2z^2 + 47376x^4z^4 + 4608x^2y^4z^2 + 4256x^2y^2z^4 + 144y^4z^4 - 65536x^6 - 65536x^4y^2 - 83968x^4z^2 - 6144x^2y^2z^2 + 65536x^4]$$

$$s2_w^{Ura} = [4x^6 - 72x^4y^2 + 324x^2y^4 + 180x^5 - 1464x^3y^2 - 1404xy^4 - 247x^4 - 1386x^2y^2 - 783y^4 - 844x^3 + 7020xy^2 - 438x^2 - 1506y^2 + 16x + 33]$$

Projection in Joint space

$$s2_j^{Ort} = [\rho_1^4\rho_2^2 + \rho_1^4\rho_3^2 + \rho_1^2\rho_2^4 + 2\rho_1^2\rho_2^2\rho_3^2 + \rho_1^2\rho_3^4 + \rho_2^4\rho_3^2 + \rho_2^2\rho_3^4 - 16\rho_1^2\rho_2^2 - 16\rho_1^2\rho_3^2 - 16\rho_2^2\rho_3^2]$$

$$s2_j^{Tri} = [\rho_1^4\rho_2^2 - 2\rho_1^4\rho_2\rho_3 + \rho_1^4\rho_3^2 - 2\rho_1^3\rho_2^3 + 2\rho_1^3\rho_2^2\rho_3 + 2\rho_1^3\rho_2\rho_3^2 - 2\rho_1^3\rho_3^3 + \rho_1^2\rho_2^4 + 2\rho_1^2\rho_2^3\rho_3 - 6\rho_1^2\rho_2^2\rho_3^2 + 2\rho_1^2\rho_2\rho_3^3 + \rho_1^2\rho_3^4 - 2\rho_1\rho_2^4\rho_3 + 2\rho_1\rho_2^3\rho_3^2 + 2\rho_1\rho_2^2\rho_3^3 - 2\rho_1\rho_2\rho_3^4 + \rho_2^4\rho_3^2 - 2\rho_2^3\rho_3^3 + \rho_2^2\rho_3^4 + \rho_1^4 - 2\rho_1^3\rho_2 - 2\rho_1^3\rho_3 + 6\rho_1^2\rho_2^2 - 6\rho_1^2\rho_2\rho_3 + 6\rho_1^2\rho_3^2 - 2\rho_1\rho_2^3 - 6\rho_1\rho_2^2\rho_3 + 12\rho_1\rho_2\rho_3^2 - 8\rho_1\rho_3^3 + \rho_2^4 - 2\rho_2^3\rho_3 + 6\rho_2^2\rho_3^2 - 8\rho_2\rho_3^3 + 4\rho_3^4 - 11\rho_1^2 - 34\rho_1\rho_2 + 56\rho_1\rho_3 - 11\rho_2^2 + 56\rho_2\rho_3 - 56\rho_3^2 + 4]$$

$$s2_j^{Hyb} = [\rho_1^4\rho_2^2 + \rho_1^4\rho_3^2 - 2\rho_1^3\rho_2^3 - 2\rho_1^3\rho_2\rho_3^2 + \rho_1^2\rho_2^4 + 2\rho_1^2\rho_2^2\rho_3^2 + \rho_1^2\rho_3^4 - 2\rho_1\rho_2^3\rho_3^2 - 2\rho_1\rho_2\rho_3^4 + \rho_2^4\rho_3^2 + \rho_2^2\rho_3^4 + \rho_1^4 - 2\rho_1^3\rho_2 + 6\rho_1^2\rho_2^2 - 10\rho_1^2\rho_3^2 - 2\rho_1\rho_2^3 + 28\rho_1\rho_2\rho_3^2 + \rho_2^4 - 10\rho_2^2\rho_3^2 + 4\rho_3^4 - 11\rho_1^2 - 34\rho_1\rho_2 - 11\rho_2^2 - 56\rho_3^2 + 4]$$

$$s2_j^{Ura} = [[4\rho_1^4\rho_2^2 - 8\rho_1^4\rho_2\rho_3 + 4\rho_1^4\rho_3^2 - 8\rho_1^3\rho_2^3 + 8\rho_1^3\rho_2^2\rho_3 + 8\rho_1^3\rho_2\rho_3^2 - 8\rho_1^3\rho_3^3 + 4\rho_1^2\rho_2^4 + 8\rho_1^2\rho_2^3\rho_3 - 24\rho_1^2\rho_2^2\rho_3^2 + 8\rho_1^2\rho_2\rho_3^3 + 4\rho_1^2\rho_3^4 - 8\rho_1\rho_2^4\rho_3 + 8\rho_1\rho_2^3\rho_3^2 + 8\rho_1\rho_2^2\rho_3^3 - 8\rho_1\rho_2\rho_3^4 + 4\rho_2^4\rho_3^2 - 8\rho_2^3\rho_3^3 + 4\rho_2^2\rho_3^4 +$$

$$4\rho_1^4 - 8\rho_1^3\rho_2 - 8\rho_1^3\rho_3 + 24\rho_1^2\rho_2^2 - 24\rho_1^2\rho_2\rho_3 + 24\rho_1^2\rho_3^2 - 20\rho_1\rho_2^3 + 12\rho_1\rho_2^2\rho_3 + 12\rho_1\rho_2\rho_3^2 - 20\rho_1\rho_3^3 + 10\rho_2^4 - 20\rho_2^3\rho_3 + 24\rho_2^2\rho_3^2 - 20\rho_2\rho_3^3 + 10\rho_3^4 - 44\rho_1^2 + 44\rho_1\rho_2 + 44\rho_1\rho_3 - 125\rho_2^2 + 206\rho_2\rho_3 - 125\rho_3^2 - 119]$$

A.3 Serial Singularities

Projection in workspace

$$s1_w^{Ort} = [(y^2 + z^2 - 4)(x^2 + z^2 - 4)(x^2 + y^2 - 4)]$$

$$s1_w^{Tri} = [(x^2 + z^2 - 2x - 3)(x^2 + z^2 + 2x - 3)(x^2 + z^2 - 4)]$$

$$s1_w^{Hyb} = [(x^2 + z^2 - 2x - 3)(x^2 + z^2 + 2x - 3)(x^2 + y^2 - 4)]$$

$$s1_w^{Ura} = [(x^2 + y^2 - 2x - 3)(40000x^2 + 40000y^2 + 40000x - 69200y - 120071)(40000x^2 + 40000y^2 + 40000x + 69200y - 120071)]$$

Projection in Joint space

$$s1_j^{Ort} = [(\rho_1^4\rho_2^2 + \rho_1^4\rho_3^2 + 4\rho_1^2\rho_2^2\rho_3^2 + \rho_2^4\rho_3^2 + \rho_2^2\rho_3^4 - 16\rho_2^2\rho_3^2)(\rho_1^4\rho_3^2 + \rho_1^2\rho_2^4 + 4\rho_1^2\rho_2^2\rho_3^2 + \rho_1^2\rho_3^4 + \rho_2^4\rho_3^2 - 16\rho_1^2\rho_3^2)(\rho_1^4\rho_2^2 + \rho_1^2\rho_2^4 + 4\rho_1^2\rho_2^2\rho_3^2 + \rho_1^2\rho_3^4 + \rho_2^2\rho_3^4 - 16\rho_1^2\rho_2^2)]$$

$$s1_j^{Tri} = [(\rho_1^2 + 2\rho_1\rho_2 - 4\rho_1\rho_3 - \rho_2^2 + 2\rho_3^2 - 2)(\rho_1^2 - 2\rho_1\rho_2 - \rho_2^2 + 4\rho_2\rho_3 - 2\rho_3^2 + 2)(\rho_1^2 - 2\rho_1\rho_3 + \rho_2^2 - 2\rho_2\rho_3 + 2\rho_3^2 + 2)]$$

$$s1_j^{Hyb} = [(\rho_1^4\rho_3^2 - 4\rho_1^3\rho_2\rho_3^2 + 6\rho_1^2\rho_2^2\rho_3^2 - 4\rho_1\rho_2^3\rho_3^2 + \rho_2^4\rho_3^2 + \rho_1^4 + 4\rho_1^3\rho_2 + 2\rho_1^2\rho_2^2 + 12\rho_1^2\rho_3^2 - 4\rho_1\rho_2^3 - 8\rho_1\rho_2\rho_3^2 + \rho_2^4 + 4\rho_2^2\rho_3^2 + 4\rho_3^4 - 4\rho_1^2 - 8\rho_1\rho_2 + 4\rho_2^2 - 56\rho_3^2 + 4)(\rho_1^4\rho_3^2 - 4\rho_1^3\rho_2\rho_3^2 + 6\rho_1^2\rho_2^2\rho_3^2 - 4\rho_1\rho_2^3\rho_3^2 + \rho_2^4\rho_3^2 + \rho_1^4 - 4\rho_1^3\rho_2 + 2\rho_1^2\rho_2^2 + 4\rho_1^2\rho_3^2 + 4\rho_1\rho_2^3 - 8\rho_1\rho_2\rho_3^2 + \rho_2^4 + 12\rho_2^2\rho_3^2 + 4\rho_3^4 + 4\rho_1^2 - 8\rho_1\rho_2 - 4\rho_2^2 - 56\rho_3^2 + 4)(\rho_1^4\rho_2^2 - 2\rho_1^3\rho_2^2 - 2\rho_1^2\rho_2\rho_3^2 + \rho_1^2\rho_2^4 + 4\rho_1^2\rho_2^2\rho_3^2 + \rho_1^2\rho_3^4 - 2\rho_1\rho_2^3\rho_3^2 - 2\rho_1\rho_2\rho_3^4 + \rho_2^2\rho_3^4 + \rho_1^4 - 2\rho_1^3\rho_2 + 6\rho_1^2\rho_2^2 + 6\rho_1^2\rho_3^2 - 2\rho_1\rho_2^3 - 4\rho_1\rho_2\rho_3^2 + \rho_2^4 + 6\rho_2^2\rho_3^2 + 4\rho_3^4 - 11\rho_1^2 - 34\rho_1\rho_2 - 11\rho_2^2 + 8\rho_3^2 + 4)]$$

$$s1_j^{Ura} := [(4\rho_1^4 - 8\rho_1^3\rho_2 - 8\rho_1^3\rho_3 + 44\rho_1^2\rho_2^2 - 64\rho_1^2\rho_2\rho_3 + 44\rho_1^2\rho_3^2 - 40\rho_1\rho_2^3 + 32\rho_1\rho_2^2\rho_3 + 32\rho_1\rho_2\rho_3^2 - 40\rho_1\rho_3^3 + 10\rho_2^4 - 16\rho_2^3\rho_3 + 10\rho_3^4 + 20\rho_1^2 - 20\rho_1\rho_2 - 20\rho_1\rho_3 + 10\rho_2^2 + 10\rho_3^2 - 119)(4\rho_1^4 - 16\rho_1^3\rho_2 + 20\rho_1^2\rho_2^2 + 8\rho_1^2\rho_2\rho_3 - 4\rho_1^2\rho_3^2 - 8\rho_1\rho_2^3 - 16\rho_1\rho_2^2\rho_3 + 8\rho_1\rho_2\rho_3^2 + 10\rho_2^4 - 32\rho_2^3\rho_3 + 56\rho_2^2\rho_3^2 - 40\rho_2\rho_3^3 + 10\rho_3^4 + 16\rho_1^2 - 32\rho_1\rho_2 + 26\rho_2^2 - 20\rho_2\rho_3 + 10\rho_3^2 - 119)(4\rho_1^4 - 16\rho_1^3\rho_3 - 4\rho_1^2\rho_2^2 + 8\rho_1^2\rho_2\rho_3 + 20\rho_1^2\rho_3^2 + 8\rho_1\rho_2^2\rho_3 - 16\rho_1\rho_2\rho_3^2 - 8\rho_1\rho_3^3 + 10\rho_2^4 - 40\rho_2^3\rho_3 + 56\rho_2^2\rho_3^2 - 32\rho_2\rho_3^3 + 10\rho_3^4 + 16\rho_1^2 - 32\rho_1\rho_3 + 10\rho_2^2 - 20\rho_2\rho_3 + 26\rho_3^2 - 119)]$$

A.4 Reconfigurable 3RPS Parallel Robot

3RPS

Record[

$$\text{Equations} = [4k^2q_1^4 + 8k^2q_1^2q_2^2 + 4k^2q_1^2q_3^2 + 4k^2q_1^2q_4^2 + 4k^2q_2^4 + 4k^2q_2^2q_3^2 + 4k^2q_2^2q_4^2 - 4k^2q_1^2 - 4k^2q_2^2 + 4kp_xq_1^2 + 4kp_xq_2^2 + 4kp_yq_1q_4 + 4kp_yq_2q_3 - 4kp_zq_1q_3 + 4kp_zq_2q_4 - 4kq_1^2 - 4kq_2^2 + k^2 - 2kp_x + p_x^2 + p_y^2 + p_z^2 - \rho_1^2 + 2k - 2p_x + 1, -2k^2q_1^2q_2q_3s_3 - 2k^2q_2^2q_3s_3 - 2k^2q_2q_3^2s_3 -$$

$2k^2q_2q_3q_4^2s_3 + 4k^2q_1^4 + 5k^2q_1^2q_2^2 + 7k^2q_1^2q_3^2 + 4k^2q_1^2q_4^2 + k^2q_2^4 + 4k^2q_2^2q_3^2 + k^2q_2^2q_4^2 + 3k^2q_3^4 + 3k^2q_3^2q_4^2 +$
 $2k^2q_2q_3s_3 - 2kp_xq_1q_4s_3 + 2kp_xq_2q_3s_3 + 2kp_yq_1^2s_3 + 2kp_yq_3^2s_3 + 2kp_zq_1q_2s_3 + 2kp_zq_3q_4s_3 -$
 $4k^2q_1^2 - k^2q_2^2 - 3k^2q_3^2 - 2kp_xq_1^2 - 2kp_xq_2^2 - 2kp_yq_1q_4 - 2kp_yq_2q_3 + 2kp_zq_1q_3 - 2kp_zq_2q_4 +$
 $2kq_2q_3s_3 - kp_ys_3 - 4kq_1^2 - kq_2^2 - 3kq_3^2 + k^2 + kp_x + p_x^2 + p_y^2 - p_ys_3 + p_z^2 - \rho_2^2 + 2k + p_x + 1,$
 $2k^2q_1^2q_2q_3s_3 + 2k^2q_2^3q_3s_3 + 2k^2q_2q_3^3s_3 + 2k^2q_2q_3q_4^2s_3 + 4k^2q_1^4 + 5k^2q_1^2q_2^2 + 7k^2q_1^2q_3^2 + 4k^2q_1^2q_4^2 +$
 $k^2q_2^4 + 4k^2q_2^2q_3^2 + k^2q_2^2q_4^2 + 3k^2q_3^4 + 3k^2q_3^2q_4^2 - 2k^2q_2q_3s_3 + 2kp_xq_1q_4s_3 - 2kp_xq_2q_3s_3 - 2kp_yq_1^2s_3 -$
 $2kp_yq_3^2s_3 - 2kp_zq_1q_2s_3 - 2kp_zq_3q_4s_3 - 4k^2q_1^2 - k^2q_2^2 - 3k^2q_3^2 - 2kp_xq_1^2 - 2kp_xq_2^2 - 2kp_yq_1q_4 -$
 $2kp_yq_2q_3 + 2kp_zq_1q_3 - 2kp_zq_2q_4 - 2kq_2q_3s_3 + kp_ys_3 - 4kq_1^2 - kq_2^2 - 3kq_3^2 + k^2 + kp_x + p_x^2 + p_y^2 + p_ys_3 +$
 $p_z^2 - \rho_3^2 + 2k + p_x + 1, 2kq_1q_4 + 2kq_2q_3 + p_y, -\sqrt{(3)kq_2^2} + \sqrt{(3)kq_3^2} - 4kq_1q_4 + 2kq_2q_3 + \sqrt{(3)p_x + p_y},$
 $\sqrt{(3)kq_2^2} - \sqrt{(3)kq_3^2} - 4kq_1q_4 + 2kq_2q_3 - \sqrt{(3)p_x + p_y}, q_1q_4, q_1^2 + q_2^2 + q_3^2 + q_4^2 - 1, s_3^2 - 3],$
Constraints = $[0 < s_3, 0 < \rho_1, 0 < \rho_2, 0 < \rho_3],$ **PoseVariables** = $[p_x, p_y, p_z, q_1, q_2, q_3,$
 $q_4],$ **ArticularVariables** = $[\rho_1, \rho_2, \rho_3],$ **PassiveVariables** = $[],$ **GeometricParameters**
= $[k, s_3],$ **GenericEquations** = $[4h^2q_1^4 + 8h^2q_1^2q_2^2 + 4h^2q_1^2q_3^2 + 4h^2q_1^2q_4^2 + 4h^2q_2^4 + 4h^2q_2^2q_3^2 +$
 $4h^2q_2^2q_4^2 - 4ghq_1^2 - 4ghq_2^2 - 4h^2q_1^2 - 4h^2q_2^2 + 4hp_xq_1^2 + 4hp_xq_2^2 + 4hp_yq_1q_4 + 4hp_yq_2q_3 - 4hp_zq_1q_3 +$
 $4hp_zq_2q_4 + g^2 + 2gh - 2gp_x + h^2 - 2hp_x + p_x^2 + p_y^2 + p_z^2 - \rho_1^2, -2h^2q_1^2q_2q_3s_3 - 2h^2q_2^3q_3s_3 -$
 $2h^2q_2q_3^3s_3 - 2h^2q_2q_3q_4^2s_3 + 4h^2q_1^4 + 5h^2q_1^2q_2^2 + 7h^2q_1^2q_3^2 + 4h^2q_1^2q_4^2 + h^2q_2^4 + 4h^2q_2^2q_3^2 + h^2q_2^2q_4^2 +$
 $3h^2q_3^4 + 3h^2q_3^2q_4^2 + 2ghq_2q_3s_3 + 2h^2q_2q_3s_3 - 2hp_xq_1q_4s_3 + 2hp_xq_2q_3s_3 + 2hp_yq_1^2s_3 + 2hp_yq_3^2s_3 +$
 $2hp_zq_1q_2s_3 + 2hp_zq_3q_4s_3 - 4ghq_1^2 - ghq_2^2 - 3ghq_3^2 - 4h^2q_1^2 - h^2q_2^2 - 3h^2q_3^2 - 2hp_xq_1^2 - 2hp_xq_2^2 -$
 $2hp_yq_1q_4 - 2hp_yq_2q_3 + 2hp_zq_1q_3 - 2hp_zq_2q_4 - gp_ys_3 - hp_ys_3 + g^2 + 2gh + gp_x + h^2 + hp_x + p_x^2 +$
 $p_y^2 + p_z^2 - \rho_2^2, 2h^2q_1^2q_2q_3s_3 + 2h^2q_2^3q_3s_3 + 2h^2q_2q_3^3s_3 + 2h^2q_2q_3q_4^2s_3 + 4h^2q_1^4 + 5h^2q_1^2q_2^2 + 7h^2q_1^2q_3^2 +$
 $4h^2q_1^2q_4^2 + h^2q_2^4 + 4h^2q_2^2q_3^2 + h^2q_2^2q_4^2 + 3h^2q_3^4 + 3h^2q_3^2q_4^2 - 2ghq_2q_3s_3 - 2h^2q_2q_3s_3 + 2hp_xq_1q_4s_3 -$
 $2hp_xq_2q_3s_3 - 2hp_yq_1^2s_3 - 2hp_yq_3^2s_3 - 2hp_zq_1q_2s_3 - 2hp_zq_3q_4s_3 - 4ghq_1^2 - ghq_2^2 - 3ghq_3^2 -$
 $4h^2q_1^2 - h^2q_2^2 - 3h^2q_3^2 - 2hp_xq_1^2 - 2hp_xq_2^2 - 2hp_yq_1q_4 - 2hp_yq_2q_3 + 2hp_zq_1q_3 - 2hp_zq_2q_4 +$
 $gp_ys_3 + hp_ys_3 + g^2 + 2gh + gp_x + h^2 + hp_x + p_x^2 + p_y^2 + p_z^2 - \rho_3^2, (2q_1q_4 + 2q_2q_3)h + p_y,$
 $p_y - (1/2(2q_1q_4 + 2q_2q_3))h + (1/2(2q_1^2 + 2q_2^2 - 1))h\sqrt{(3)} + \sqrt{(3)p_x} - (1/2)\sqrt{(3)}(2q_1^2 + 2q_2^2 -$
 $1)h + (3/2(-2q_1q_4 + 2q_2q_3))h, p_y - (1/2(2q_1q_4 + 2q_2q_3))h - (1/2(2q_1^2 + 2q_2^2 - 1))h\sqrt{(3)} - \sqrt{(3)p_x} +$
 $(1/2)\sqrt{(3)}(2q_1^2 + 2q_2^2 - 1)h + (3/2(-2q_1q_4 + 2q_2q_3))h, q_1q_4, q_1^2 + q_2^2 + q_3^2 + q_4^2 - 1, s_3^2 - 3],$
GenericConstraints = $[0 < s_3, 0 < \rho_1, 0 < \rho_2, 0 < \rho_3],$ **Precision** = 4, **PoseValues**
= $[],$ **ArticularValues** = $[],$ **PassiveValues** = $[],$ **GeometricValues** = $[g = 1, h = k],$
DefaultPlotRanges = $[],$ **Points** = $[R1 = [g, 0, 0], R2 = [-(1/2)g, (1/2)g\sqrt{(3)}, 0], R3 =$
 $[-(1/2)g, -(1/2)g\sqrt{(3)}, 0], E1 = [h(2q_1^2 + 2q_2^2 - 1) + p_x, (2q_1q_4 + 2q_2q_3)h + p_y, h(-2q_1q_3 +$
 $2q_2q_4) + p_z], E2 = [p_x - (1/2)h(2q_1^2 + 2q_2^2 - 1) + (1/2(-2q_1q_4 + 2q_2q_3))h\sqrt{(3)}, p_y - (1/2(2q_1q_4 +$
 $2q_2q_3))h + (1/2(2q_1^2 + 2q_2^2 - 1))h\sqrt{(3)}, p_z - (1/2)h(-2q_1q_3 + 2q_2q_4) + (1/2(2q_1q_2 + 2q_3q_4))h\sqrt{(3)}],$
 $E3 = [p_x - (1/2)h(2q_1^2 + 2q_2^2 - 1) - (1/2(-2q_1q_4 + 2q_2q_3))h\sqrt{(3)}, p_y - (1/2(2q_1q_4 + 2q_2q_3))h -$
 $(1/2(2q_1^2 + 2q_2^2 - 1))h\sqrt{(3)}, p_z - (1/2)h(-2q_1q_3 + 2q_2q_4) - (1/2(2q_1q_2 + 2q_3q_4))h\sqrt{(3)}], P =$
 $[p_x, p_y, p_z],$ **Loops** = $[[R1, R2, R3], [E1, E2, E3], [E1, P], [E2, P], [E3, P]],$ **Chains** = $[],$
Actuators = $[[R1, E1], [R2, E2], [R3, E3]],$ **Model** = "3RPS"
]

Parallel Singularities: Projection in (p_x, p_y, p_z) for OM^1

$$[36864p_x^16p_y^2 + 16384p_x^16p_z^2 + 196608p_x^14p_y^4 + 131072p_x^14p_y^2p_z^2 + 16384p_x^14p_z^4 + 409600p_x^12p_y^6 + 458752p_x^12p_y^4p_z^2 + 40960p_x^12p_y^2p_z^4 - 32768p_x^12p_z^6 + 393216p_x^10p_y^8 + 917504p_x^10p_y^6p_z^2 + 1425408p_x^10p_y^4p_z^4 +$$

$$\begin{aligned}
& 393216p_x^1 0p_y^2 p_z^6 - 32768p_x^1 0p_z^8 + 122880p_x^8 p_y^1 0 + 1146880p_x^8 p_y^8 p_z^2 - 122880p_x^8 p_y^6 p_z^4 + 884736p_x^8 p_y^4 p_z^6 + \\
& 462848p_x^8 p_y^2 p_z^8 + 16384p_x^8 p_z^1 0 - 65536p_x^6 p_y^1 2 + 917504p_x^6 p_y^1 0p_z^2 - 344064p_x^6 p_y^8 p_z^4 + 507904p_x^6 p_y^4 p_z^8 + \\
& 65536p_x^6 p_y^2 p_z^1 0 + 16384p_x^6 p_z^2 2 - 49152p_x^4 p_y^1 4 + 458752p_x^4 p_y^1 2p_z^2 + 1105920p_x^4 p_y^1 0p_z^4 - 884736p_x^4 p_y^8 p_z^6 - \\
& 466944p_x^4 p_y^6 p_z^8 + 98304p_x^4 p_y^4 p_z^1 0 + 49152p_x^4 p_y^2 p_z^2 2 + 131072p_x^2 p_y^1 4p_z^2 - 49152p_x^2 p_y^1 2p_z^4 - 393216p_x^2 p_y^1 0p_z^6 - \\
& 442368p_x^2 p_y^8 p_z^8 + 65536p_x^2 p_y^6 p_z^1 0 + 49152p_x^2 p_y^4 p_z^2 2 + 4096p_y^8 + 16384p_y^6 p_z^2 + 24576p_y^4 p_z^4 + 32768p_y^2 p_z^6 + \\
& 36864p_y^1 0p_z^8 + 16384p_y^8 p_z^1 0 + 16384p_y^6 p_z^2 2 - 32768p_x^1 5p_z^2 - 614400p_x^1 3p_y^2 p_z^2 - 229376p_x^1 3p_z^4 + \\
& 442368p_x^1 1p_y^4 p_z^2 - 458752p_x^1 1p_y^2 p_z^4 - 163840p_x^1 1p_z^6 + 3858432p_x^9 p_y^6 p_z^2 + 1146880p_x^9 p_y^4 p_z^4 - 827392p_x^9 p_y^2 p_z^6 + \\
& 229376p_x^9 p_z^8 + 3538944p_x^7 p_y^8 p_z^2 + 4587520p_x^7 p_y^6 p_z^4 + 2752512p_x^7 p_y^4 p_z^6 + 196608p_x^7 p_z^1 0 + 417792p_x^5 p_y^1 0p_z^2 + \\
& 5734400p_x^5 p_y^8 p_z^4 + 3325952p_x^5 p_y^6 p_z^6 - 1376256p_x^5 p_y^4 p_z^8 - 196608p_x^5 p_y^2 p_z^1 0 - 16384p_x^3 p_y^1 2p_z^2 + 3211264p_x^3 p_y^1 0p_z^4 + \\
& 622592p_x^3 p_y^8 p_z^6 - 1835008p_x^3 p_y^6 p_z^8 - 983040p_x^3 p_y^4 p_z^1 0 + 270336p_x p_y^1 4p_z^2 + 688128p_x p_y^1 2p_z^4 + 712704p_x p_y^1 0p_z^6 - \\
& 688128p_x p_y^8 p_z^8 - 589824p_x p_y^6 p_z^1 0 - 1142784p_x^1 4p_y^2 - 515072p_x^1 4p_z^2 - 4952064p_x^1 2p_y^4 - 2683904p_x^1 2p_y^2 p_z^2 + \\
& 280576p_x^1 2p_z^4 - 7745536p_x^1 0p_y^6 - 7744512p_x^1 0p_y^4 p_z^2 - 1394688p_x^1 0p_y^2 p_z^4 + 890880p_x^1 0p_z^6 - 4444160p_x^8 p_y^8 - \\
& 14853120p_x^8 p_y^6 p_z^2 - 5959680p_x^8 p_y^4 p_z^4 + 3790848p_x^8 p_y^2 p_z^6 + 577536p_x^8 p_z^8 + 634880p_x^6 p_y^1 0 - 17617920p_x^6 p_y^8 p_z^2 + \\
& 6172672p_x^6 p_y^6 p_z^4 + 8024064p_x^6 p_y^4 p_z^6 + 2291712p_x^6 p_y^2 p_z^8 + 427008p_x^6 p_z^1 0 + 1396736p_x^4 p_y^1 2 - 11738112p_x^4 p_y^1 0p_z^2 + \\
& 4270080p_x^4 p_y^8 p_z^4 + 9056256p_x^4 p_y^6 p_z^6 + 3459072p_x^4 p_y^4 p_z^8 + 1281024p_x^4 p_y^2 p_z^1 0 - 55296p_x^4 p_z^2 2 + 126976p_x^2 p_y^1 4 - \\
& 3810304p_x^2 p_y^1 2p_z^2 + 3151872p_x^2 p_y^1 0p_z^4 + 4749312p_x^2 p_y^8 p_z^6 + 2320384p_x^2 p_y^6 p_z^8 + 1281024p_x^2 p_y^4 p_z^1 0 - \\
& 110592p_x^2 p_y^2 p_z^2 2 - 126976p_y^6 - 412672p_y^4 p_z^2 - 98304p_y^1 2p_z^4 + 817152p_y^1 0p_z^6 + 575488p_y^8 p_z^8 + \\
& 427008p_y^6 p_z^1 0 - 55296p_y^4 p_z^2 2 + 868864p_x^1 3p_z^2 + 5313536p_x^1 1p_y^2 p_z^2 + 870400p_x^1 1p_z^4 - 10304000p_x^9 p_y^4 p_z^2 + \\
& 2934784p_x^9 p_y^2 p_z^4 + 2984960p_x^9 p_z^6 - 32475136p_x^7 p_y^6 p_z^2 - 10727424p_x^7 p_y^4 p_z^4 + 110592p_x^7 p_y^2 p_z^6 + 1218560p_x^7 p_z^8 - \\
& 20926976p_x^5 p_y^8 p_z^2 - 15396864p_x^5 p_y^6 p_z^4 - 18315264p_x^5 p_y^4 p_z^6 - 1218560p_x^5 p_y^2 p_z^8 - 93696p_x^5 p_z^1 0 - 6999040p_x^3 p_y^1 0p_z^2 - \\
& 5904384p_x^3 p_y^8 p_z^4 - 23646208p_x^3 p_y^6 p_z^6 - 6092800p_x^3 p_y^4 p_z^8 + 187392p_x^3 p_y^2 p_z^1 0 - 3798528p_x p_y^1 2p_z^2 - \\
& 3299328p_x p_y^1 0p_z^4 - 8991744p_x p_y^8 p_z^6 - 3655680p_x p_y^6 p_z^8 + 281088p_x p_y^4 p_z^1 0 + 11957760p_x^1 2p_y^2 + 5219584p_x^1 2p_z^2 + \\
& 39859200p_x^1 0p_y^4 + 25253376p_x^1 0p_y^2 p_z^2 + 2489920p_x^1 0p_z^4 + 41187840p_x^8 p_y^6 + 64144128p_x^8 p_y^4 p_z^2 + 33641792p_x^8 p_y^2 p_z^4 - \\
& 1249280p_x^8 p_z^6 + 5314560p_x^6 p_y^8 + 97653760p_x^6 p_y^6 p_z^2 + 53155456p_x^6 p_y^4 p_z^4 + 3914752p_x^6 p_y^2 p_z^6 + 2113408p_x^6 p_z^8 - \\
& 11957760p_x^4 p_y^1 0 + 82336512p_x^4 p_y^8 p_z^2 + 20189824p_x^4 p_y^6 p_z^4 - 4525056p_x^4 p_y^4 p_z^6 + 6722688p_x^4 p_y^2 p_z^8 - \\
& 619776p_x^4 p_z^1 0 - 2657280p_x^2 p_y^1 2 + 33338880p_x^2 p_y^1 0p_z^2 + 3030848p_x^2 p_y^8 p_z^4 - 9948160p_x^2 p_y^6 p_z^6 + 6085248p_x^2 p_y^4 p_z^8 - \\
& 1239552p_x^2 p_y^2 p_z^1 0 + 46656p_x^2 p_z^2 2 + 1328640p_y^4 + 4545792p_y^1 2p_z^2 + 4844608p_y^1 0p_z^4 - 259072p_y^8 p_z^6 + \\
& 2155904p_y^6 p_z^8 - 619776p_y^4 p_z^1 0 + 46656p_y^2 p_z^2 2 - 7318400p_x^1 1p_z^2 - 10041728p_x^9 p_y^2 p_z^2 + 3398656p_x^9 p_z^4 + \\
& 51172608p_x^7 p_y^4 p_z^2 - 4644864p_x^7 p_y^2 p_z^4 - 4038528p_x^7 p_z^6 + 106693888p_x^5 p_y^6 p_z^2 - 3360768p_x^5 p_y^4 p_z^4 + 4038528p_x^5 p_y^2 p_z^6 - \\
& 1036800p_x^5 p_z^8 + 75660928p_x^3 p_y^8 p_z^2 - 36995072p_x^3 p_y^6 p_z^4 + 20192640p_x^3 p_y^4 p_z^6 + 2073600p_x^3 p_y^2 p_z^8 - 6912p_x^3 p_z^1 0 + \\
& 22862976p_x p_y^1 0p_z^2 - 8647680p_x p_y^8 p_z^4 + 12115584p_x p_y^6 p_z^6 + 3110400p_x p_y^4 p_z^8 + 20736p_x p_y^2 p_z^1 0 - \\
& 50100480p_x^1 0p_y^2 - 18447680p_x^1 0p_z^2 - 116901120p_x^8 p_y^4 - 64680256p_x^8 p_y^2 p_z^2 - 15076400p_x^8 p_z^4 - 55667200p_x^6 p_y^6 - \\
& 147732608p_x^6 p_y^4 p_z^2 - 75298880p_x^6 p_y^2 p_z^4 + 4904640p_x^6 p_z^6 + 33400320p_x^4 p_y^8 - 190600832p_x^4 p_y^6 p_z^2 - \\
& 95456160p_x^4 p_y^4 p_z^4 + 13939776p_x^4 p_y^2 p_z^6 - 3207168p_x^4 p_z^8 + 16700160p_x^2 p_y^1 0 - 104486464p_x^2 p_y^8 p_z^2 - \\
& 51976000p_x^2 p_y^6 p_z^4 + 15230016p_x^2 p_y^4 p_z^6 - 6414336p_x^2 p_y^2 p_z^8 + 212544p_x^2 p_z^1 0 - 5566720p_y^1 2 - 15385664p_y^1 0p_z^2 - \\
& 16742320p_y^8 p_z^4 + 4818624p_y^6 p_z^6 - 3207168p_y^4 p_z^8 + 212544p_y^2 p_z^1 0 - 11664p_z^2 2 + 23255776p_x^9 p_z^2 - \\
& 6096384p_x^7 p_y^2 p_z^2 - 13997696p_x^7 p_z^4 - 117181248p_x^5 p_y^4 p_z^2 + 13997696p_x^5 p_y^2 p_z^4 + 2161824p_x^5 p_z^6 - 198916352p_x^3 p_y^6 p_z^2 + \\
& 69988480p_x^3 p_y^4 p_z^4 - 4323648p_x^3 p_y^2 p_z^6 + 290304p_x^3 p_z^8 - 67735200p_x p_y^8 p_z^2 + 41993088p_x p_y^6 p_z^4 - 6485472p_x p_y^4 p_z^6 - \\
& 870912p_x p_y^2 p_z^8 + 96719760p_x^8 p_y^2 + 26364976p_x^8 p_z^2 + 128959680p_x^6 p_y^4 + 44689600p_x^6 p_y^2 p_z^2 + 23108908p_x^6 p_z^4 - \\
& 21493280p_x^4 p_y^6 + 137933088p_x^4 p_y^4 p_z^2 + 84567684p_x^4 p_y^2 p_z^4 - 4284432p_x^4 p_z^6 - 42986560p_x^2 p_y^8 + 139221184p_x^2 p_y^6 p_z^2 + \\
& 59166084p_x^2 p_y^4 p_z^4 - 8568864p_x^2 p_y^2 p_z^6 + 887112p_x^2 p_z^8 + 10746640p_y^1 0 + 19612720p_y^8 p_z^2 + 24802348p_y^6 p_z^4 - \\
& 4284432p_y^4 p_z^6 + 887112p_y^2 p_z^8 - 30772968p_x^7 p_z^2 + 30772968p_x^5 p_y^2 p_z^2 + 10253376p_x^5 p_z^4 + 153864840p_x^3 p_y^4 p_z^2 -
\end{aligned}$$

$$\begin{aligned}
& 20506752p_x^3p_y^2p_z^4 + 381024p_x^3p_z^6 + 92318904p_xp_y^6p_z^2 - 30760128p_xp_y^4p_z^4 - 1143072p_xp_y^2p_z^6 - 85476384p_x^6p_y^2 - \\
& 13128192p_x^6p_z^2 - 28492128p_x^4p_y^4 + 3290112p_x^4p_y^2p_z^2 - 12168153p_x^4p_z^4 + 47486880p_x^2p_y^6 - 67834368p_x^2p_y^4p_z^2 - \\
& 24336306p_x^2p_y^2p_z^4 - 9497376p_y^8 - 8386560p_y^6p_z^2 - 12168153p_y^4p_z^4 + 14489496p_x^5p_z^2 - 28978992p_x^3p_y^2p_z^2 - \\
& 43468488p_xp_y^4p_z^2 + 28005264p_x^4p_y^2 - 18670176p_x^2p_y^4 + 3111696p_y^6]
\end{aligned}$$

Parallel Singularities: Projection in (p_x, p_y, p_z) for OM^2

$$\begin{aligned}
& [k^10p_x^4p_z^4 + 2k^10p_x^2p_y^2p_z^4 + k^10p_y^4p_z^4 + 4k^9p_x^7p_z^2 - 4k^9p_x^5p_y^2p_z^2 - 20k^9p_x^3p_y^4p_z^2 - 12k^9p_xp_y^6p_z^2 - 36k^8p_x^8p_y^2 - \\
& 12k^8p_x^8p_z^2 - 48k^8p_x^6p_y^4 - 48k^8p_x^6p_y^2p_z^2 - 11k^8p_x^6p_z^4 + 8k^8p_x^4p_y^6 - 72k^8p_x^4p_y^4p_z^2 - 33k^8p_x^4p_y^2p_z^4 + 4k^8p_x^4p_z^6 + \\
& 16k^8p_x^2p_y^8 - 48k^8p_x^2p_y^6p_z^2 - 33k^8p_x^2p_y^4p_z^4 + 8k^8p_x^2p_y^2p_z^6 - 2k^8p_x^2p_z^8 - 4k^8p_y^10 - 12k^8p_y^8p_z^2 - 11k^8p_y^6p_z^4 + \\
& 4k^8p_y^4p_z^6 - 2k^8p_y^2p_z^8 - 12k^7p_x^9p_z^2 + 16k^7p_x^7p_z^4 + 72k^7p_x^5p_y^4p_z^2 - 16k^7p_x^5p_y^2p_z^4 - 4k^7p_x^5p_z^6 + 96k^7p_x^3p_y^6p_z^2 - \\
& 80k^7p_x^3p_y^4p_z^4 + 8k^7p_x^3p_y^2p_z^6 + 36k^7p_xp_y^8p_z^2 - 48k^7p_xp_y^6p_z^4 + 12k^7p_xp_y^4p_z^6 + 144k^6p_x^10p_y^2 + 52k^6p_x^10p_z^2 + \\
& 336k^6p_x^8p_y^4 + 116k^6p_x^8p_y^2p_z^2 + 11k^6p_x^8p_z^4 + 160k^6p_x^6p_y^6 + 328k^6p_x^6p_y^4p_z^2 - 28k^6p_x^6p_y^2p_z^4 - 12k^6p_x^6p_z^6 - \\
& 96k^6p_x^4p_y^8 + 552k^6p_x^4p_y^6p_z^2 + 42k^6p_x^4p_y^4p_z^4 - 36k^6p_x^4p_y^2p_z^6 + 16k^6p_x^4p_z^8 - 48k^6p_x^2p_y^10 + 324k^6p_x^2p_y^8p_z^2 + \\
& 84k^6p_x^2p_y^6p_z^4 - 36k^6p_x^2p_y^4p_z^6 + 32k^6p_x^2p_y^2p_z^8 - 4k^6p_x^2p_z^10 + 16k^6p_y^12 + 36k^6p_y^10p_z^2 + 3k^6p_y^8p_z^4 - \\
& 12k^6p_y^6p_z^6 + 16k^6p_y^4p_z^8 - 4k^6p_y^2p_z^10 + k^6p_z^12 + 12k^5p_x^11p_z^2 - 132k^5p_x^9p_y^2p_z^2 - 64k^5p_x^9p_z^4 + 312k^5p_x^7p_y^4p_z^2 - \\
& 36k^5p_x^7p_z^6 + 56k^5p_x^5p_y^6p_z^2 + 384k^5p_x^5p_y^4p_z^4 + 36k^5p_x^5p_y^2p_z^6 - 16k^5p_x^5p_z^8 - 388k^5p_x^3p_y^8p_z^2 + 512k^5p_x^3p_y^6p_z^4 + \\
& 180k^5p_x^3p_y^4p_z^6 + 32k^5p_x^3p_y^2p_z^8 + 8k^5p_x^3p_z^10 + 12k^5p_xp_y^10p_z^2 + 192k^5p_xp_y^8p_z^4 + 108k^5p_xp_y^6p_z^6 + 48k^5p_xp_y^4p_z^8 - \\
& 24k^5p_xp_y^2p_z^10 - 216k^4p_x^12p_y^2 - 84k^4p_x^12p_z^2 - 720k^4p_x^10p_y^4 - 216k^4p_x^10p_y^2p_z^2 + 23k^4p_x^10p_z^4 - 744k^4p_x^8p_y^6 - \\
& 588k^4p_x^8p_y^4p_z^2 + 187k^4p_x^8p_y^2p_z^4 + 64k^4p_x^8p_z^6 - 96k^4p_x^6p_y^8 - 1360k^4p_x^6p_y^6p_z^2 + 326k^4p_x^6p_y^4p_z^4 + 400k^4p_x^6p_y^2p_z^6 + \\
& 34k^4p_x^6p_z^8 + 216k^4p_x^4p_y^10 - 1452k^4p_x^4p_y^8p_z^2 + 214k^4p_x^4p_y^6p_z^4 + 432k^4p_x^4p_y^4p_z^6 - 114k^4p_x^4p_y^2p_z^8 + 20k^4p_x^4p_z^10 + \\
& 48k^4p_x^2p_y^12 - 600k^4p_x^2p_y^10p_z^2 + 83k^4p_x^2p_y^8p_z^4 + 176k^4p_x^2p_y^6p_z^6 + 246k^4p_x^2p_y^4p_z^8 + 40k^4p_x^2p_y^2p_z^10 - \\
& 9k^4p_x^2p_z^12 - 24k^4p_y^14 - 52k^4p_y^12p_z^2 + 31k^4p_y^10p_z^4 + 80k^4p_y^8p_z^6 + 10k^4p_y^6p_z^8 + 20k^4p_y^4p_z^10 - 9k^4p_y^2p_z^12 - \\
& 4k^3p_x^13p_z^2 + 280k^3p_x^11p_y^2p_z^2 + 112k^3p_x^11p_z^4 - 460k^3p_x^9p_y^4p_z^2 + 112k^3p_x^9p_y^2p_z^4 + 104k^3p_x^9p_z^6 - 1136k^3p_x^7p_y^6p_z^2 - \\
& 672k^3p_x^7p_y^4p_z^4 - 288k^3p_x^7p_y^2p_z^6 - 48k^3p_x^7p_z^8 + 164k^3p_x^5p_y^8p_z^2 - 1568k^3p_x^5p_y^6p_z^4 + 432k^3p_x^5p_y^4p_z^6 + \\
& 48k^3p_x^5p_y^2p_z^8 - 36k^3p_x^5p_z^10 + 472k^3p_x^3p_y^10p_z^2 - 1232k^3p_x^3p_y^8p_z^4 - 1440k^3p_x^3p_y^6p_z^6 + 240k^3p_x^3p_y^4p_z^8 + \\
& 72k^3p_x^3p_y^2p_z^10 - 84k^3p_xp_y^12p_z^2 - 336k^3p_xp_y^10p_z^4 - 216k^3p_xp_y^8p_z^6 + 144k^3p_xp_y^6p_z^8 + 108k^3p_xp_y^4p_z^10 + \\
& 144k^2p_x^14p_y^2 + 60k^2p_x^14p_z^2 + 624k^2p_x^12p_y^4 + 276k^2p_x^12p_y^2p_z^2 - 8k^2p_x^12p_z^4 + 976k^2p_x^10p_y^6 + 780k^2p_x^10p_y^4p_z^2 - \\
& 120k^2p_x^10p_y^2p_z^4 - 88k^2p_x^10p_z^6 + 560k^2p_x^8p_y^8 + 1604k^2p_x^8p_y^6p_z^2 + 1008k^2p_x^8p_y^4p_z^4 - 152k^2p_x^8p_y^2p_z^6 - \\
& 80k^2p_x^8p_z^8 - 80k^2p_x^6p_y^10 + 2036k^2p_x^6p_y^8p_z^2 - 1968k^2p_x^6p_y^6p_z^4 - 496k^2p_x^6p_y^4p_z^6 + 40k^2p_x^6p_y^2p_z^8 - 36k^2p_x^6p_z^10 - \\
& 176k^2p_x^4p_y^12 + 1404k^2p_x^4p_y^10p_z^2 + 792k^2p_x^4p_y^8p_z^4 - 944k^2p_x^4p_y^6p_z^6 - 360k^2p_x^4p_y^4p_z^8 - 108k^2p_x^4p_y^2p_z^10 + \\
& 24k^2p_x^4p_z^12 - 16k^2p_x^2p_y^14 + 452k^2p_x^2p_y^12p_z^2 - 216k^2p_x^2p_y^10p_z^4 - 568k^2p_x^2p_y^8p_z^6 - 520k^2p_x^2p_y^6p_z^8 - \\
& 108k^2p_x^2p_y^4p_z^10 + 48k^2p_x^2p_y^2p_z^12 + 16k^2p_y^16 + 44k^2p_y^14p_z^2 - 56k^2p_y^12p_z^4 - 40k^2p_y^10p_z^6 - 36k^2p_y^8p_z^8 + \\
& 24k^2p_y^6p_z^10 - 144k^2p_x^13p_y^2p_z^2 - 64k^2p_x^13p_z^4 + 96k^2p_x^11p_y^4p_z^2 - 128k^2p_x^11p_y^2p_z^4 - 64k^2p_x^11p_z^6 + 848k^2p_x^9p_y^6p_z^2 + \\
& 320k^2p_x^9p_y^4p_z^4 + 80k^2p_x^9p_y^2p_z^6 + 64k^2p_x^9p_z^8 + 576k^2p_x^7p_y^8p_z^2 + 1280k^2p_x^7p_y^6p_z^4 + 64k^2p_x^7p_z^10 - 240k^2p_x^5p_y^10p_z^2 + \\
& 1600k^2p_x^5p_y^8p_z^4 + 672k^2p_x^5p_y^6p_z^6 - 384k^2p_x^5p_y^4p_z^8 - 64k^2p_x^5p_y^2p_z^10 - 160k^2p_x^3p_y^12p_z^2 + 896k^2p_x^3p_y^10p_z^4 + 960k^2p_x^3p_y^8p_z^6 - \\
& 512k^2p_x^3p_y^6p_z^8 - 320k^2p_x^3p_y^4p_z^10 + 48k^2p_xp_y^14p_z^2 + 192k^2p_xp_y^12p_z^4 + 144k^2p_xp_y^10p_z^6 - 192k^2p_xp_y^8p_z^8 - \\
& 192k^2p_xp_y^6p_z^10 - 36p_x^16p_y^2 - 16p_x^16p_z^2 - 192p_x^14p_y^4 - 128p_x^14p_y^2p_z^2 - 16p_x^14p_z^4 - 400p_x^12p_y^6 - 448p_x^12p_y^4p_z^2 - \\
& 40p_x^12p_y^2p_z^4 + 32p_x^12p_z^6 - 384p_x^10p_y^8 - 896p_x^10p_y^6p_z^2 - 1392p_x^10p_y^4p_z^4 - 384p_x^10p_y^2p_z^6 + 32p_x^10p_z^8 - \\
& 120p_x^8p_y^10 - 1120p_x^8p_y^8p_z^2 + 120p_x^8p_y^6p_z^4 - 864p_x^8p_y^4p_z^6 - 452p_x^8p_y^2p_z^8 - 16p_x^8p_z^10 + 64p_x^6p_y^12 - 896p_x^6p_y^10p_z^2 + \\
& 336p_x^6p_y^8p_z^4 - 496p_x^6p_y^6p_z^6 - 64p_x^6p_y^4p_z^8 - 16p_x^6p_z^10 + 48p_x^4p_y^14 - 448p_x^4p_y^12p_z^2 - 1080p_x^4p_y^10p_z^4 + \\
& 864p_x^4p_y^8p_z^6 + 456p_x^4p_y^6p_z^8 - 96p_x^4p_y^4p_z^10 - 48p_x^4p_y^2p_z^12 - 128p_x^2p_y^14p_z^2 + 48p_x^2p_y^12p_z^4 + 384p_x^2p_y^10p_z^6 + \\
& 432p_x^2p_y^8p_z^8 - 64p_x^2p_y^6p_z^10 - 48p_x^2p_y^4p_z^12 - 4p_y^18 - 16p_y^16p_z^2 - 24p_y^14p_z^4 - 32p_y^12p_z^6 - 36p_y^10p_z^8 -
\end{aligned}$$

$$\begin{aligned}
& 16p_y^8p_z^10 - 16p_y^6p_z^12 - 8k^9p_x^6p_z^2 - 24k^9p_x^4p_y^2p_z^2 - 8k^9p_x^4p_z^4 - 24k^9p_x^2p_y^4p_z^2 - 16k^9p_x^2p_y^2p_z^4 - 8k^9p_y^6p_z^2 - \\
& 8k^9p_y^4p_z^4 - 8k^8p_x^7p_z^2 + 8k^8p_x^5p_y^2p_z^2 + 40k^8p_x^3p_y^4p_z^2 + 24k^8p_xp_y^6p_z^2 + 288k^7p_x^8p_y^2 + 120k^7p_x^8p_z^2 + \\
& 384k^7p_x^6p_y^4 + 480k^7p_x^6p_y^2p_z^2 + 48k^7p_x^6p_z^4 - 64k^7p_x^6p_y^4 + 720k^7p_x^4p_y^4p_z^2 + 144k^7p_x^4p_y^2p_z^4 + 8k^7p_x^4p_z^6 - \\
& 128k^7p_x^2p_y^8 + 480k^7p_x^2p_y^6p_z^2 + 144k^7p_x^2p_y^4p_z^4 + 16k^7p_x^2p_y^2p_z^6 + 8k^7p_x^2p_z^8 + 32k^7p_y^10 + 120k^7p_y^8p_z^2 + \\
& 48k^7p_y^6p_z^4 + 8k^7p_y^4p_z^6 + 8k^7p_y^2p_z^8 - 40k^6p_x^9p_z^2 + 32k^6p_x^7p_z^4 + 240k^6p_x^5p_y^4p_z^2 - 32k^6p_x^5p_y^2p_z^4 - 24k^6p_x^5p_z^6 + \\
& 320k^6p_x^3p_y^6p_z^2 - 160k^6p_x^3p_y^4p_z^4 + 48k^6p_x^3p_y^2p_z^6 - 16k^6p_x^3p_z^8 + 120k^6p_xp_y^8p_z^2 - 96k^6p_xp_y^6p_z^4 + 72k^6p_xp_y^4p_z^6 + \\
& 48k^6p_xp_y^2p_z^8 - 864k^5p_x^10p_y^2 - 344k^5p_x^10p_z^2 - 2016k^5p_x^8p_y^4 - 1432k^5p_x^8p_y^2p_z^2 - 72k^5p_x^8p_z^4 - 960k^5p_x^6p_y^6 - \\
& 3056k^5p_x^6p_y^4p_z^2 - 1152k^5p_x^6p_y^2p_z^4 - 88k^5p_x^6p_z^6 + 576k^5p_x^4p_y^8 - 3504k^5p_x^4p_y^6p_z^2 - 720k^5p_x^4p_y^4p_z^4 + \\
& 312k^5p_x^4p_y^2p_z^6 - 24k^5p_x^4p_z^8 + 288k^5p_x^2p_y^10 - 1848k^5p_x^2p_y^8p_z^2 + 192k^5p_x^2p_y^6p_z^4 - 648k^5p_x^2p_y^4p_z^6 - 48k^5p_x^2p_y^2p_z^8 - \\
& 96k^5p_y^12 - 312k^5p_y^10p_z^2 - 168k^5p_y^8p_z^4 - 24k^5p_y^6p_z^6 - 24k^5p_y^4p_z^8 + 136k^4p_x^11p_z^2 + 424k^4p_x^9p_y^2p_z^2 - \\
& 64k^4p_x^9p_z^4 - 1584k^4p_x^7p_y^4p_z^2 + 576k^4p_x^7p_y^2p_z^4 + 360k^4p_x^7p_z^6 - 2352k^4p_x^5p_y^6p_z^2 - 1728k^4p_x^5p_y^4p_z^4 - \\
& 360k^4p_x^5p_y^2p_z^6 + 32k^4p_x^5p_z^8 - 984k^4p_x^3p_y^8p_z^2 + 1728k^4p_x^3p_y^6p_z^4 - 1800k^4p_x^3p_y^4p_z^6 - 64k^4p_x^3p_y^2p_z^8 + \\
& 16k^4p_x^3p_z^10 - 504k^4p_xp_y^10p_z^2 - 1080k^4p_xp_y^8p_z^2 - 96k^4p_xp_y^6p_z^4 - 48k^4p_xp_y^4p_z^6 + 864k^3p_x^12p_y^2 + \\
& 360k^3p_x^12p_z^2 + 2880k^3p_x^10p_y^4 + 1584k^3p_x^10p_y^2p_z^2 - 96k^3p_x^10p_z^4 + 2976k^3p_x^8p_y^6 + 4056k^3p_x^8p_y^4p_z^2 + \\
& 1248k^3p_x^8p_y^2p_z^4 - 176k^3p_x^8p_z^6 + 384k^3p_x^6p_y^8 + 6560k^3p_x^6p_y^6p_z^2 + 1344k^3p_x^6p_y^4p_z^4 - 704k^3p_x^6p_y^2p_z^6 + \\
& 32k^3p_x^6p_z^8 - 864k^3p_x^4p_y^10 + 5784k^3p_x^4p_y^8p_z^2 - 1344k^3p_x^4p_y^6p_z^4 - 1056k^3p_x^4p_y^4p_z^6 - 480k^3p_x^4p_y^2p_z^8 - \\
& 56k^3p_x^4p_z^10 - 192k^3p_x^2p_y^12 + 2352k^3p_x^2p_y^10p_z^2 - 1248k^3p_x^2p_y^8p_z^4 - 704k^3p_x^2p_y^6p_z^6 + 480k^3p_x^2p_y^4p_z^8 - \\
& 112k^3p_x^2p_y^2p_z^10 + 96k^3p_x^14 + 296k^3p_y^12p_z^2 + 96k^3p_y^10p_z^4 - 176k^3p_y^8p_z^6 - 32k^3p_y^6p_z^8 - 56k^3p_y^4p_z^10 - \\
& 120k^2p_x^13p_z^2 - 816k^2p_x^11p_y^2p_z^2 - 96k^2p_x^11p_z^4 + 1560k^2p_x^9p_y^4p_z^2 - 672k^2p_x^9p_y^2p_z^4 - 400k^2p_x^9p_z^6 + 4832k^2p_x^7p_y^6p_z^2 + \\
& 2112k^2p_x^7p_y^4p_z^4 - 576k^2p_x^7p_y^2p_z^6 - 32k^2p_x^7p_z^8 + 2872k^2p_x^5p_y^8p_z^2 + 2240k^2p_x^5p_y^6p_z^4 + 4512k^2p_x^5p_y^4p_z^6 + \\
& 32k^2p_x^5p_y^2p_z^8 + 8k^2p_x^5p_z^10 + 848k^2p_x^3p_y^10p_z^2 + 32k^2p_x^3p_y^8p_z^4 + 1984k^2p_x^3p_y^6p_z^6 + 160k^2p_x^3p_y^4p_z^8 - 16k^2p_x^3p_y^2p_z^10 + \\
& 552k^2p_xp_y^12p_z^2 + 480k^2p_xp_y^10p_z^4 + 1392k^2p_xp_y^8p_z^4 + 96k^2p_xp_y^6p_z^6 - 24k^2p_xp_y^4p_z^8 - 288k^2p_x^14p_y^2 - \\
& 128k^2p_x^14p_z^2 - 1248k^2p_x^12p_y^4 - 608k^2p_x^12p_y^2p_z^2 + 128k^2p_x^12p_z^4 - 1952k^2p_x^10p_y^6 - 1728k^2p_x^10p_y^4p_z^2 - 96k^2p_x^10p_y^2p_z^4 + \\
& 256k^2p_x^10p_z^6 - 1120k^2p_x^8p_y^8 - 3488k^2p_x^8p_y^6p_z^2 - 96k^2p_x^8p_y^4p_z^4 + 1568k^2p_x^8p_y^2p_z^6 + 128k^2p_x^8p_z^8 + 160k^2p_x^6p_y^10 - \\
& 4352k^2p_x^6p_y^8p_z^2 + 1600k^2p_x^6p_y^6p_z^4 + 2944k^2p_x^6p_y^4p_z^6 + 1664k^2p_x^6p_y^2p_z^8 + 128k^2p_x^6p_z^10 + 352k^2p_x^4p_y^12 - 2976k^2p_x^4p_y^10p_z^2 + \\
& 2496k^2p_x^4p_y^8p_z^4 + 2496k^2p_x^4p_y^6p_z^6 + 1152k^2p_x^4p_y^4p_z^8 + 384k^2p_x^4p_y^2p_z^10 + 32k^2p_x^2p_y^14 - 960k^2p_x^2p_y^12p_z^2 + 1056k^2p_x^2p_y^10p_z^4 + \\
& 1152k^2p_x^2p_y^8p_z^6 - 128k^2p_x^2p_y^6p_z^8 + 384k^2p_x^2p_y^4p_z^10 - 32k^2p_y^16 - 96k^2p_y^14p_z^2 + 32k^2p_y^12p_z^4 + 288k^2p_y^10p_z^6 + \\
& 256k^2p_y^8p_z^8 + 128k^2p_y^6p_z^10 + 32p_x^15p_z^2 + 384p_x^13p_y^2p_z^2 + 128p_x^13p_z^4 - 288p_x^11p_y^4p_z^2 + 256p_x^11p_y^2p_z^4 + \\
& 64p_x^11p_z^6 - 2496p_x^9p_y^6p_z^2 - 640p_x^9p_y^4p_z^4 + 928p_x^9p_y^2p_z^6 - 128p_x^9p_z^8 - 2592p_x^7p_y^8p_z^2 - 2560p_x^7p_y^6p_z^4 - \\
& 2688p_x^7p_y^4p_z^6 - 96p_x^7p_z^8 - 768p_x^5p_y^10p_z^2 - 3200p_x^5p_y^8p_z^4 - 2240p_x^5p_y^6p_z^6 + 768p_x^5p_y^4p_z^8 + 96p_x^5p_y^2p_z^10 - \\
& 224p_x^3p_y^12p_z^2 - 1792p_x^3p_y^10p_z^4 + 832p_x^3p_y^8p_z^6 + 1024p_x^3p_y^6p_z^8 + 480p_x^3p_y^4p_z^10 - 192p_xp_y^14p_z^2 - 384p_xp_y^12p_z^4 - \\
& 480p_xp_y^10p_z^6 + 384p_xp_y^8p_z^8 + 288p_xp_y^6p_z^10 + 4k^9p_x^5p_z^2 - 8k^9p_x^3p_y^2p_z^2 - 12k^9p_xp_y^4p_z^2 + 72k^8p_x^6p_y^2 + \\
& 52k^8p_x^6p_z^2 + 24k^8p_x^4p_y^4 + 156k^8p_x^4p_y^2p_z^2 + 32k^8p_x^4p_z^4 - 40k^8p_x^2p_y^6 + 156k^8p_x^2p_y^4p_z^2 + 64k^8p_x^2p_y^2p_z^4 + \\
& 8k^8p_y^8 + 52k^8p_y^6p_z^2 + 32k^8p_y^4p_z^4 - 76k^7p_x^7p_z^2 + 76k^7p_x^5p_y^2p_z^2 + 24k^7p_x^5p_z^4 + 380k^7p_x^3p_y^4p_z^2 - 48k^7p_x^3p_y^2p_z^4 + \\
& 4k^7p_x^3p_z^6 + 228k^7p_xp_y^6p_z^2 - 72k^7p_xp_y^4p_z^4 - 12k^7p_xp_y^2p_z^6 - 1152k^6p_x^8p_y^2 - 428k^6p_x^8p_z^2 - 1536k^6p_x^6p_y^4 - \\
& 1136k^6p_x^6p_y^2p_z^2 - 208k^6p_x^6p_z^4 + 256k^6p_x^4p_y^6 - 2376k^6p_x^4p_y^4p_z^2 - 1344k^6p_x^4p_y^2p_z^4 - 20k^6p_x^4p_z^6 + 512k^6p_x^2p_y^8 - \\
& 2032k^6p_x^2p_y^6p_z^2 - 144k^6p_x^2p_y^4p_z^4 - 40k^6p_x^2p_y^2p_z^6 - 24k^6p_x^2p_z^8 - 128k^6p_y^10 - 364k^6p_y^8p_z^2 - 288k^6p_y^6p_z^4 - \\
& 20k^6p_y^4p_z^6 - 24k^6p_y^2p_z^8 + 428k^5p_x^9p_z^2 - 144k^5p_x^7p_y^2p_z^2 - 304k^5p_x^7p_z^4 - 2040k^5p_x^5p_y^4p_z^2 + 304k^5p_x^5p_y^2p_z^4 + \\
& 204k^5p_x^5p_z^6 - 3728k^5p_x^3p_y^6p_z^2 + 1520k^5p_x^3p_y^4p_z^4 - 408k^5p_x^3p_y^2p_z^6 + 24k^5p_x^3p_z^8 - 1236k^5p_xp_y^8p_z^2 + \\
& 912k^5p_xp_y^6p_z^4 - 612k^5p_xp_y^4p_z^6 - 72k^5p_xp_y^2p_z^8 + 2448k^4p_x^10p_y^2 + 940k^4p_x^10p_z^2 + 5712k^4p_x^8p_y^4 + 3980k^4p_x^8p_y^2p_z^2 + \\
& 720k^4p_x^8p_z^4 + 2720k^4p_x^6p_y^6 + 8440k^4p_x^6p_y^4p_z^2 + 5328k^4p_x^6p_y^2p_z^4 - 84k^4p_x^6p_z^6 - 1632k^4p_x^4p_y^8 + 9560k^4p_x^4p_y^6p_z^2 + \\
& 5136k^4p_x^4p_y^4p_z^4 + 36k^4p_x^4p_y^2p_z^6 + 152k^4p_x^4p_z^8 - 816k^4p_x^2p_y^10 + 5020k^4p_x^2p_y^8p_z^2 + 1520k^4p_x^2p_y^6p_z^4 -
\end{aligned}$$

$$\begin{aligned}
& 444k^4 p_x^2 p_y^4 p_z^6 + 304k^4 p_x^2 p_y^2 p_z^8 - 32k^4 p_x^2 p_z^{10} + 272k^4 p_y^2 + 860k^4 p_y^0 p_z^2 + 992k^4 p_y^8 p_z^4 - 52k^4 p_y^6 p_z^6 + \\
& 152k^4 p_y^4 p_z^8 - 32k^4 p_y^2 p_z^{10} - 580k^3 p_x^1 p_z^2 - 868k^3 p_x^9 p_y^2 p_z^2 + 280k^3 p_x^9 p_z^4 + 4248k^3 p_x^7 p_y^4 p_z^2 - 864k^3 p_x^7 p_y^2 p_z^4 - \\
& 832k^3 p_x^7 p_z^6 + 8568k^3 p_x^5 p_y^6 p_z^2 + 1488k^3 p_x^5 p_y^4 p_z^4 + 832k^3 p_x^5 p_y^2 p_z^6 - 152k^3 p_x^5 p_z^8 + 5868k^3 p_x^3 p_y^8 p_z^2 - \\
& 4064k^3 p_x^3 p_y^6 p_z^4 + 4160k^3 p_x^3 p_y^4 p_z^6 + 304k^3 p_x^3 p_y^2 p_z^8 + 4k^3 p_x^3 p_z^{10} + 1836k^3 p_x p_y^0 p_z^2 - 552k^3 p_x p_y^8 p_z^4 + \\
& 2496k^3 p_x p_y^6 p_z^6 + 456k^3 p_x p_y^4 p_z^8 - 12k^3 p_x p_y^2 p_z^{10} - 1728k^2 p_x^2 p_y^2 - 740k^2 p_x^2 p_z^2 - 5760k^2 p_x^0 p_y^4 - \\
& 4152k^2 p_x^0 p_y^2 p_z^2 - 480k^2 p_x^0 p_z^4 - 5952k^2 p_x^8 p_y^6 - 10428k^2 p_x^8 p_y^4 p_z^2 - 6576k^2 p_x^8 p_y^2 p_z^4 + 248k^2 p_x^8 p_z^6 - \\
& 768k^2 p_x^6 p_y^8 - 14480k^2 p_x^6 p_y^6 p_z^2 - 10368k^2 p_x^6 p_y^4 p_z^4 - 1600k^2 p_x^6 p_y^2 p_z^6 - 320k^2 p_x^6 p_z^8 + 1728k^2 p_x^4 p_y^{10} - \\
& 11292k^2 p_x^4 p_y^8 p_z^2 - 3872k^2 p_x^4 p_y^6 p_z^4 + 624k^2 p_x^4 p_y^4 p_z^6 - 1680k^2 p_x^4 p_y^2 p_z^8 + 140k^2 p_x^4 p_z^{10} + 384k^2 p_x^2 p_y^{12} - \\
& 4536k^2 p_x^2 p_y^{10} p_z^2 - 544k^2 p_x^2 p_y^8 p_z^4 + 2432k^2 p_x^2 p_y^6 p_z^6 - 480k^2 p_x^2 p_y^4 p_z^8 + 280k^2 p_x^2 p_y^2 p_z^{10} - 192k^2 p_y^{14} - \\
& 708k^2 p_y^2 p_z^2 - 944k^2 p_y^0 p_z^4 - 40k^2 p_y^8 p_z^6 - 400k^2 p_y^6 p_z^8 + 140k^2 p_y^4 p_z^{10} + 224k p_x^1 p_z^2 + 880k p_x^1 p_y^2 p_z^2 - \\
& 1840k p_x^9 p_y^4 p_z^2 + 864k p_x^9 p_y^2 p_z^4 + 768k p_x^9 p_z^6 - 6304k p_x^7 p_y^6 p_z^2 - 2304k p_x^7 p_y^4 p_z^4 - 144k p_x^7 p_y^2 p_z^6 + 512k p_x^7 p_z^8 - \\
& 5504k p_x^5 p_y^8 p_z^2 - 1344k p_x^5 p_y^6 p_z^4 - 4080k p_x^5 p_y^4 p_z^6 - 512k p_x^5 p_y^2 p_z^8 + 32k p_x^5 p_z^{10} - 2512k p_x^3 p_y^0 p_z^2 + \\
& 1536k p_x^3 p_y^8 p_z^4 - 6448k p_x^3 p_y^6 p_z^6 - 2560k p_x^3 p_y^4 p_z^8 - 64k p_x^3 p_y^2 p_z^{10} - 816k p_x p_y^2 p_z^2 - 288k p_x p_y^0 p_z^4 - \\
& 2256k p_x p_y^8 p_z^6 - 1536k p_x p_y^6 p_z^8 - 96k p_x p_y^4 p_z^{10} + 360p_x^1 4p_y^2 + 176p_x^1 4p_z^2 + 1560p_x^1 2p_y^4 + 1088p_x^1 2p_y^2 p_z^2 - \\
& 64p_x^1 2p_z^4 + 2440p_x^0 p_y^6 + 3216p_x^0 p_y^4 p_z^2 + 1488p_x^0 p_y^2 p_z^4 - 288p_x^0 p_z^6 + 1400p_x^8 p_y^8 + 5664p_x^8 p_y^6 p_z^2 + \\
& 3408p_x^8 p_y^4 p_z^4 - 1008p_x^8 p_y^2 p_z^6 - 192p_x^8 p_z^8 - 200p_x^6 p_y^{10} + 6096p_x^6 p_y^8 p_z^2 + 800p_x^6 p_y^6 p_z^4 - 2304p_x^6 p_y^4 p_z^6 + \\
& 168p_x^6 p_y^2 p_z^8 - 144p_x^6 p_z^{10} - 440p_x^4 p_y^2 + 3840p_x^4 p_y^0 p_z^2 - 2208p_x^4 p_y^8 p_z^4 - 2976p_x^4 p_y^6 p_z^6 - 840p_x^4 p_y^4 p_z^8 - \\
& 432p_x^4 p_y^2 p_z^{10} - 40p_x^2 p_y^4 + 1264p_x^2 p_y^2 p_z^2 - 1008p_x^2 p_y^0 p_z^4 - 1632p_x^2 p_y^8 p_z^6 - 1288p_x^2 p_y^6 p_z^8 - 432p_x^2 p_y^4 p_z^{10} + \\
& 40p_y^6 + 160p_y^4 p_z^2 + 144p_y^2 p_z^4 - 240p_y^0 p_z^6 - 88p_y^8 p_z^8 - 144p_y^6 p_z^{10} - 32k^8 p_x^5 p_z^2 + 64k^8 p_x^3 p_y^2 p_z^2 + \\
& 96k^8 p_x p_y^4 p_z^2 - 576k^7 p_x^6 p_y^2 - 160k^7 p_x^6 p_z^2 - 192k^7 p_x^4 p_y^4 - 192k^7 p_x^4 p_y^2 p_z^2 - 80k^7 p_x^4 p_z^4 + 320k^7 p_x^2 p_y^6 - \\
& 672k^7 p_x^2 p_y^4 p_z^2 - 160k^7 p_x^2 p_y^2 p_z^4 - 64k^7 p_y^8 - 128k^7 p_y^6 p_z^2 - 80k^7 p_y^4 p_z^4 + 448k^6 p_x^7 p_z^2 - 448k^6 p_x^5 p_y^2 p_z^2 - \\
& 192k^6 p_x^5 p_z^4 - 2240k^6 p_x^3 p_y^4 p_z^2 + 384k^6 p_x^3 p_y^2 p_z^4 - 16k^6 p_x^3 p_z^6 - 1344k^6 p_x p_y^6 p_z^2 + 576k^6 p_x p_y^4 p_z^4 + 48k^6 p_x p_y^2 p_z^6 + \\
& 2880k^5 p_x^8 p_y^2 + 832k^5 p_x^8 p_z^2 + 3840k^5 p_x^6 p_y^4 + 1024k^5 p_x^6 p_y^2 p_z^2 + 608k^5 p_x^6 p_z^4 - 640k^5 p_x^4 p_y^6 + 4224k^5 p_x^4 p_y^4 p_z^2 + \\
& 3840k^5 p_x^4 p_y^2 p_z^4 + 32k^5 p_x^4 p_z^6 - 1280k^5 p_x^2 p_y^8 + 4608k^5 p_x^2 p_y^6 p_z^2 + 480k^5 p_x^2 p_y^4 p_z^4 + 64k^5 p_x^2 p_y^2 p_z^6 + 32k^5 p_x^2 p_z^8 + \\
& 320k^5 p_y^{10} + 576k^5 p_y^8 p_z^2 + 832k^5 p_y^6 p_z^4 + 32k^5 p_y^4 p_z^6 + 32k^5 p_y^2 p_z^8 - 1312k^4 p_x^9 p_z^2 + 576k^4 p_x^7 p_y^2 p_z^2 + \\
& 928k^4 p_x^7 p_z^4 + 5760k^4 p_x^5 p_y^4 p_z^2 - 928k^4 p_x^5 p_y^2 p_z^4 - 272k^4 p_x^5 p_z^6 + 11712k^4 p_x^3 p_y^6 p_z^2 - 4640k^4 p_x^3 p_y^4 p_z^4 + \\
& 544k^4 p_x^3 p_y^2 p_z^6 + 16k^4 p_x^3 p_z^8 + 3744k^4 p_x p_y^8 p_z^2 - 2784k^4 p_x p_y^6 p_z^4 + 816k^4 p_x p_y^4 p_z^6 - 48k^4 p_x p_y^2 p_z^8 - \\
& 4032k^3 p_x^0 p_y^2 - 1440k^3 p_x^0 p_z^2 - 9408k^3 p_x^8 p_y^4 - 4608k^3 p_x^8 p_y^2 p_z^2 - 1424k^3 p_x^8 p_z^4 - 4480k^3 p_x^6 p_y^6 - \\
& 10944k^3 p_x^6 p_y^4 p_z^2 - 8000k^3 p_x^6 p_y^2 p_z^4 + 448k^3 p_x^6 p_z^6 + 2688k^3 p_x^4 p_y^8 - 14976k^3 p_x^4 p_y^6 p_z^2 - 9312k^3 p_x^4 p_y^4 p_z^4 - \\
& 672k^3 p_x^4 p_y^2 p_z^6 - 336k^3 p_x^4 p_z^8 + 1344k^3 p_x^2 p_y^{10} - 8352k^3 p_x^2 p_y^8 p_z^2 - 4416k^3 p_x^2 p_y^6 p_z^4 + 2688k^3 p_x^2 p_y^4 p_z^6 - \\
& 672k^3 p_x^2 p_y^2 p_z^8 - 448k^3 p_y^{12} - 1152k^3 p_y^0 p_z^2 - 1680k^3 p_y^8 p_z^4 + 224k^3 p_y^6 p_z^6 - 336k^3 p_y^4 p_z^8 + 1152k^2 p_x^1 p_z^2 + \\
& 1152k^2 p_x^9 p_y^2 p_z^2 - 480k^2 p_x^9 p_z^4 - 6912k^2 p_x^7 p_y^4 p_z^2 - 576k^2 p_x^7 p_y^2 p_z^4 + 320k^2 p_x^7 p_z^6 - 16128k^2 p_x^5 p_y^6 p_z^2 + \\
& 4992k^2 p_x^5 p_y^4 p_z^4 - 320k^2 p_x^5 p_y^2 p_z^6 + 160k^2 p_x^5 p_z^8 - 12672k^2 p_x^3 p_y^8 p_z^2 + 2624k^2 p_x^3 p_y^6 p_z^4 - 1600k^2 p_x^3 p_y^4 p_z^6 - \\
& 320k^2 p_x^3 p_y^2 p_z^8 - 3456k^2 p_x p_y^{10} p_z^2 + 1632k^2 p_x p_y^8 p_z^4 - 960k^2 p_x p_y^6 p_z^6 - 480k^2 p_x p_y^4 p_z^8 + 1728k p_x^1 2p_y^2 + \\
& 768k p_x^1 2p_z^2 + 5760k p_x^0 p_y^4 + 4032k p_x^0 p_y^2 p_z^2 + 896k p_x^0 p_z^4 + 5952k p_x^8 p_y^6 + 10176k p_x^8 p_y^4 p_z^2 + 7072k p_x^8 p_y^2 p_z^4 + \\
& 768k p_x^6 p_y^8 + 14720k p_x^6 p_y^6 p_z^2 + 12416k p_x^6 p_y^4 p_z^4 + 3456k p_x^6 p_y^2 p_z^6 + 640k p_x^6 p_z^8 - 1728k p_x^4 p_y^{10} + 11904k p_x^4 p_y^8 p_z^2 + \\
& 8384k p_x^4 p_y^6 p_z^4 + 1152k p_x^4 p_y^4 p_z^6 + 1632k p_x^4 p_y^2 p_z^8 - 384k p_x^2 p_y^{12} + 4800k p_x^2 p_y^{10} p_z^2 + 3328k p_x^2 p_y^8 p_z^4 - \\
& 1920k p_x^2 p_y^6 p_z^6 + 2112k p_x^2 p_y^4 p_z^8 + 192k p_y^{14} + 704k p_y^{12} p_z^2 + 1184k p_y^{10} p_z^4 + 384k p_y^8 p_z^6 + 608k p_y^6 p_z^8 - \\
& 256p_x^1 3p_z^2 - 1088p_x^1 1p_y^2 p_z^2 - 256p_x^1 1p_z^4 + 2240p_x^9 p_y^4 p_z^2 + 320p_x^9 p_y^2 p_z^4 - 512p_x^9 p_z^6 + 7552p_x^7 p_y^6 p_z^2 - \\
& 512p_x^7 p_z^8 + 6272p_x^5 p_y^8 p_z^2 + 2688p_x^5 p_y^6 p_z^4 + 3072p_x^5 p_y^4 p_z^6 + 512p_x^5 p_y^2 p_z^8 + 2752p_x^3 p_y^{10} p_z^2 + 3840p_x^3 p_y^8 p_z^4 + \\
& 4096p_x^3 p_y^6 p_z^6 + 2560p_x^3 p_y^4 p_z^8 + 960p_x p_y^{12} p_z^2 + 576p_x p_y^{10} p_z^4 + 1536p_x p_y^8 p_z^6 + 1536p_x p_y^6 p_z^8 - 36k^8 p_x^4 p_y^2 + \\
& 24k^8 p_x^2 p_y^4 - 4k^8 p_y^6 + 128k^7 p_x^5 p_z^2 - 256k^7 p_x^3 p_y^2 p_z^2 - 384k^7 p_x p_y^4 p_z^2 + 1872k^6 p_x^6 p_y^2 + 288k^6 p_x^6 p_z^2 +
\end{aligned}$$

$$\begin{aligned}
& 624k^6 p_x^4 p_y^4 - 720k^6 p_x^4 p_y^2 p_z^2 + 176k^6 p_x^4 p_z^4 - 1040k^6 p_x^2 p_y^6 + 1920k^6 p_x^2 p_y^4 p_z^2 + 352k^6 p_x^2 p_y^2 p_z^4 + 208k^6 p_y^8 + \\
& 112k^6 p_y^6 p_z^2 + 176k^6 p_y^4 p_z^4 - 1120k^5 p_x^7 p_z^2 + 1120k^5 p_x^5 p_y^2 p_z^2 + 512k^5 p_x^5 p_z^4 + 5600k^5 p_x^3 p_y^4 p_z^2 - 1024k^5 p_x^3 p_y^2 p_z^4 + \\
& 16k^5 p_x^3 p_z^6 + 3360k^5 p_x p_y^6 p_z^2 - 1536k^5 p_x p_y^4 p_z^4 - 48k^5 p_x p_y^2 p_z^6 - 4824k^4 p_x^8 p_y^2 - 1088k^4 p_x^8 p_z^2 - 6432k^4 p_x^6 p_y^4 - \\
& 752k^4 p_x^6 p_y^2 p_z^2 - 1392k^4 p_x^6 p_z^4 + 1072k^4 p_x^4 p_y^6 - 5328k^4 p_x^4 p_y^4 p_z^2 - 3816k^4 p_x^4 p_y^2 p_z^4 + 272k^4 p_x^4 p_z^6 + \\
& 2144k^4 p_x^2 p_y^8 - 6352k^4 p_x^2 p_y^6 p_z^2 - 4416k^4 p_x^2 p_y^4 p_z^4 + 544k^4 p_x^2 p_y^2 p_z^6 - 32k^4 p_x^2 p_z^8 - 536k^4 p_y^{10} - 688k^4 p_y^8 p_z^2 - \\
& 1352k^4 p_y^6 p_z^4 + 272k^4 p_y^4 p_z^6 - 32k^4 p_y^2 p_z^8 + 1984k^3 p_x^9 p_z^2 - 576k^3 p_x^7 p_y^2 p_z^2 - 1312k^3 p_x^7 p_z^4 - 9792k^3 p_x^5 p_y^4 p_z^2 + \\
& 1312k^3 p_x^5 p_y^2 p_z^4 - 32k^3 p_x^5 p_z^6 - 17088k^3 p_x^3 p_y^6 p_z^2 + 6560k^3 p_x^3 p_y^4 p_z^4 + 64k^3 p_x^3 p_y^2 p_z^6 - 5760k^3 p_x p_y^8 p_z^2 + \\
& 3936k^3 p_x p_y^6 p_z^4 + 96k^3 p_x p_y^4 p_z^6 + 4176k^2 p_x^10 p_y^2 + 1440k^2 p_x^10 p_z^2 + 9744k^2 p_x^8 p_y^4 + 4176k^2 p_x^8 p_y^2 p_z^2 + \\
& 1664k^2 p_x^8 p_z^4 + 4640k^2 p_x^6 p_y^6 + 10368k^2 p_x^6 p_y^4 p_z^2 + 4784k^2 p_x^6 p_y^2 p_z^4 - 608k^2 p_x^6 p_z^6 - 2784k^2 p_x^4 p_y^8 + \\
& 15072k^2 p_x^4 p_y^6 p_z^2 + 9360k^2 p_x^4 p_y^4 p_z^4 - 3408k^2 p_x^4 p_y^2 p_z^6 + 384k^2 p_x^4 p_z^8 - 1392k^2 p_x^2 p_y^{10} + 8544k^2 p_x^2 p_y^8 p_z^2 + \\
& 7696k^2 p_x^2 p_y^6 p_z^4 - 768k^2 p_x^2 p_y^4 p_z^6 + 768k^2 p_x^2 p_y^2 p_z^8 + 464k^2 p_y^{12} + 1104k^2 p_y^{10} p_z^2 + 1456k^2 p_y^8 p_z^4 - 784k^2 p_y^6 p_z^6 + \\
& 384k^2 p_y^4 p_z^8 - 992k p_x^11 p_z^2 - 416k p_x^9 p_y^2 p_z^2 + 800k p_x^9 p_z^4 + 4416k p_x^7 p_y^4 p_z^2 + 608k p_x^7 p_z^6 + 12992k p_x^5 p_y^6 p_z^2 - \\
& 4800k p_x^5 p_y^4 p_z^4 - 608k p_x^5 p_y^2 p_z^6 + 96k p_x^5 p_z^8 + 11936k p_x^3 p_y^8 p_z^2 - 6400k p_x^3 p_y^6 p_z^4 - 3040k p_x^3 p_y^4 p_z^6 - 192k p_x^3 p_y^2 p_z^8 + \\
& 2784k p_x p_y^{10} p_z^2 - 2400k p_x p_y^8 p_z^4 - 1824k p_x p_y^6 p_z^6 - 288k p_x p_y^4 p_z^8 - 1188p_x^{12} p_y^2 - 640p_x^{12} p_z^2 - 3960p_x^{10} p_y^4 - \\
& 2832p_x^{10} p_y^2 p_z^2 - 448p_x^{10} p_z^4 - 4092p_x^8 p_y^6 - 7248p_x^8 p_y^4 p_z^2 - 4184p_x^8 p_y^2 p_z^4 - 256p_x^8 p_z^6 - 528p_x^6 p_y^8 - \\
& 11680p_x^6 p_y^6 p_z^2 - 7072p_x^6 p_y^4 p_z^4 + 560p_x^6 p_y^2 p_z^6 - 448p_x^6 p_z^8 + 1188p_x^4 p_y^{10} - 10272p_x^4 p_y^8 p_z^2 - 4048p_x^4 p_y^6 p_z^4 - \\
& 1008p_x^4 p_y^4 p_z^6 - 1380p_x^4 p_y^2 p_z^8 + 264p_x^2 p_y^{12} - 4176p_x^2 p_y^{10} p_z^2 - 1376p_x^2 p_y^8 p_z^4 - 1904p_x^2 p_y^6 p_z^6 - 1320p_x^2 p_y^4 p_z^8 - \\
& 132p_y^{14} - 528p_y^{12} p_z^2 - 664p_y^{10} p_z^4 - 80p_y^8 p_z^6 - 452p_y^6 p_z^8 + 288k^7 p_x^4 p_y^2 - 192k^7 p_x^2 p_y^4 + 32k^7 p_y^6 - \\
& 320k^6 p_x^5 p_z^2 + 640k^6 p_x^3 p_y^2 p_z^2 + 960k^6 p_x p_y^4 p_z^2 - 3168k^5 p_x^6 p_y^2 - 320k^5 p_x^6 p_z^2 - 1056k^5 p_x^4 p_y^4 + 1632k^5 p_x^4 p_y^2 p_z^2 - \\
& 320k^5 p_x^4 p_z^4 + 1760k^5 p_x^2 p_y^6 - 2688k^5 p_x^2 p_y^4 p_z^2 - 640k^5 p_x^2 p_y^2 p_z^4 - 352k^5 p_y^8 - 32k^5 p_y^6 p_z^2 - 320k^5 p_y^4 p_z^4 + \\
& 1664k^4 p_x^7 p_z^2 - 1664k^4 p_x^5 p_y^2 p_z^2 - 640k^4 p_x^5 p_z^4 - 8320k^4 p_x^3 p_y^4 p_z^2 + 1280k^4 p_x^3 p_y^2 p_z^4 - 4992k^4 p_x p_y^6 p_z^2 + \\
& 1920k^4 p_x p_y^4 p_z^4 + 5472k^3 p_x^8 p_y^2 + 1152k^3 p_x^8 p_z^2 + 7296k^3 p_x^6 p_y^4 + 576k^3 p_x^6 p_y^2 p_z^2 + 1472k^3 p_x^6 p_z^4 - 1216k^3 p_x^4 p_y^6 + \\
& 5568k^3 p_x^4 p_y^4 p_z^2 + 1824k^3 p_x^4 p_y^2 p_z^4 - 640k^3 p_x^4 p_z^6 - 2432k^3 p_x^2 p_y^8 + 6848k^3 p_x^2 p_y^6 p_z^2 + 6144k^3 p_x^2 p_y^4 p_z^4 - \\
& 1280k^3 p_x^2 p_y^2 p_z^6 + 608k^3 p_y^{10} + 704k^3 p_y^8 p_z^2 + 1184k^3 p_y^6 p_z^4 - 640k^3 p_y^4 p_z^6 - 1856k^2 p_x^9 p_z^2 + 896k^2 p_x^7 p_z^4 + \\
& 11136k^2 p_x^5 p_y^4 p_z^2 - 896k^2 p_x^5 p_y^2 p_z^4 + 448k^2 p_x^5 p_z^6 + 14848k^2 p_x^3 p_y^6 p_z^2 - 4480k^2 p_x^3 p_y^4 p_z^4 - 896k^2 p_x^3 p_y^2 p_z^6 + \\
& 5568k^2 p_x p_y^8 p_z^2 - 2688k^2 p_x p_y^6 p_z^4 - 1344k^2 p_x p_y^4 p_z^6 - 2592k p_x^{10} p_y^2 - 832k p_x^{10} p_z^2 - 6048k p_x^8 p_y^4 - \\
& 2720k p_x^8 p_y^2 p_z^2 - 1152k p_x^8 p_z^4 - 2880k p_x^6 p_y^6 - 6400k p_x^6 p_y^4 p_z^2 + 288k p_x^6 p_y^2 p_z^4 + 960k p_x^6 p_z^6 + 1728k p_x^4 p_y^8 - \\
& 8640k p_x^4 p_y^6 p_z^2 - 5280k p_x^4 p_y^4 p_z^4 + 2592k p_x^4 p_y^2 p_z^6 + 864k p_x^2 p_y^{10} - 4800k p_x^2 p_y^8 p_z^2 - 7328k p_x^2 p_y^6 p_z^4 + \\
& 3072k p_x^2 p_y^4 p_z^6 - 288k p_y^{12} - 672k p_y^{10} p_z^2 - 608k p_y^8 p_z^4 + 928k p_y^6 p_z^6 + 512p_x^{11} p_z^2 + 512p_x^9 p_y^2 p_z^2 - \\
& 256p_x^9 p_z^4 - 3072p_x^7 p_y^4 p_z^2 - 768p_x^7 p_z^6 - 7168p_x^5 p_y^6 p_z^2 + 1536p_x^5 p_y^4 p_z^4 + 768p_x^5 p_y^2 p_z^6 - 5632p_x^3 p_y^8 p_z^2 + \\
& 2048p_x^3 p_y^6 p_z^4 + 3840p_x^3 p_y^4 p_z^6 - 1536p_x p_y^{10} p_z^2 + 768p_x p_y^8 p_z^4 + 2304p_x p_y^6 p_z^6 - 864k^6 p_x^4 p_y^2 + 576k^6 p_x^2 p_y^4 - \\
& 96k^6 p_y^6 + 512k^5 p_x^5 p_z^2 - 1024k^5 p_x^3 p_y^2 p_z^2 - 1536k^5 p_x p_y^4 p_z^2 + 3168k^4 p_x^6 p_y^2 + 256k^4 p_x^6 p_z^2 + 1056k^4 p_x^4 p_y^4 + \\
& 192k^4 p_x^4 p_y^2 p_z^2 + 512k^4 p_x^4 p_z^4 - 1760k^4 p_x^2 p_y^6 + 1152k^4 p_x^2 p_y^4 p_z^2 + 1024k^4 p_x^2 p_y^2 p_z^4 + 352k^4 p_y^8 + 192k^4 p_y^6 p_z^2 + \\
& 512k^4 p_y^4 p_z^4 - 1472k^3 p_x^7 p_z^2 + 1472k^3 p_x^5 p_y^2 p_z^2 + 256k^3 p_x^5 p_z^4 + 7360k^3 p_x^3 p_y^4 p_z^2 - 512k^3 p_x^3 p_y^2 p_z^4 + 4416k^3 p_x p_y^6 p_z^2 - \\
& 768k^3 p_x p_y^4 p_z^4 - 3744k^2 p_x^8 p_y^2 - 960k^2 p_x^8 p_z^2 - 4992k^2 p_x^6 p_y^4 - 1536k^2 p_x^6 p_y^2 p_z^2 - 640k^2 p_x^6 p_z^4 + 832k^2 p_x^4 p_y^6 - \\
& 4992k^2 p_x^4 p_y^4 p_z^2 - 2784k^2 p_x^4 p_y^2 p_z^4 + 512k^2 p_x^4 p_z^6 + 1664k^2 p_x^2 p_y^8 - 5120k^2 p_x^2 p_y^6 p_z^2 - 1344k^2 p_x^2 p_y^4 p_z^4 + \\
& 1024k^2 p_x^2 p_y^2 p_z^6 - 416k^2 p_y^{10} - 704k^2 p_y^8 p_z^2 - 736k^2 p_y^6 p_z^4 + 512k^2 p_y^4 p_z^6 + 960k p_x^9 p_z^2 - 256k p_x^7 p_z^4 - \\
& 5760k p_x^5 p_y^4 p_z^2 + 256k p_x^5 p_y^2 p_z^4 + 64k p_x^5 p_z^6 - 7680k p_x^3 p_y^6 p_z^2 + 1280k p_x^3 p_y^4 p_z^4 - 128k p_x^3 p_y^2 p_z^6 - 2880k p_x p_y^8 p_z^2 + \\
& 768k p_x p_y^6 p_z^4 - 192k p_x p_y^4 p_z^6 + 1440p_x^{10} p_y^2 + 704p_x^{10} p_z^2 + 3360p_x^8 p_y^4 + 1792p_x^8 p_y^2 p_z^2 + 128p_x^8 p_z^4 + \\
& 1600p_x^6 p_y^6 + 4736p_x^6 p_y^4 p_z^2 + 800p_x^6 p_y^2 p_z^4 - 576p_x^6 p_z^6 - 960p_x^4 p_y^8 + 7424p_x^4 p_y^6 p_z^2 + 864p_x^4 p_y^4 p_z^4 - 1728p_x^4 p_y^2 p_z^6 - \\
& 480p_x^2 p_y^{10} + 4288p_x^2 p_y^8 p_z^2 + 352p_x^2 p_y^6 p_z^4 - 1728p_x^2 p_y^4 p_z^6 + 160p_y^{12} + 512p_y^{10} p_z^2 + 160p_y^8 p_z^4 - 576p_y^6 p_z^6 + \\
& 1152k^5 p_x^4 p_y^2 - 768k^5 p_x^2 p_y^4 + 128k^5 p_y^6 - 512k^4 p_x^5 p_z^2 + 1024k^4 p_x^3 p_y^2 p_z^2 + 1536k^4 p_x p_y^4 p_z^2 - 2304k^3 p_x^6 p_y^2 -
\end{aligned}$$

$$\begin{aligned}
& 256k^3p_x^6p_z^2 - 768k^3p_x^4p_y^4 - 1920k^3p_x^4p_y^2p_z^2 - 512k^3p_x^4p_z^4 + 1280k^3p_x^2p_y^6 - 1024k^3p_x^2p_y^2p_z^4 - 256k^3p_y^8 - \\
& 384k^3p_y^6p_z^2 - 512k^3p_y^4p_z^4 + 768k^2p_x^7p_z^2 - 768k^2p_x^5p_y^2p_z^2 + 256k^2p_x^5p_z^4 - 3840k^2p_x^3p_y^4p_z^2 - 512k^2p_x^3p_y^2p_z^4 - \\
& 2304k^2p_x^2p_y^6p_z^2 - 768k^2p_x^2p_y^4p_z^4 + 1152k^2p_x^8p_y^2 + 256k^2p_x^8p_z^2 + 1536k^2p_x^6p_y^4 + 2176k^2p_x^6p_y^2p_z^2 + 512k^2p_x^6p_z^4 - \\
& 256k^2p_x^4p_y^6 + 1920k^2p_x^4p_y^4p_z^2 + 1536k^2p_x^4p_y^2p_z^4 - 512k^2p_x^2p_y^8 + 384k^2p_x^2p_y^6p_z^2 + 1536k^2p_x^2p_y^4p_z^4 + 128k^2p_y^{10} + \\
& 384k^2p_y^8p_z^2 + 512k^2p_y^6p_z^4 - 256k^2p_x^9p_z^2 - 256k^2p_x^7p_z^4 + 1536k^2p_x^5p_y^4p_z^2 + 256k^2p_x^5p_y^2p_z^4 + 2048k^2p_x^3p_y^6p_z^2 + 1280k^2p_x^3p_y^4p_z^4 + \\
& 768k^2p_x^3p_y^2p_z^6 + 768k^2p_x^2p_y^6p_z^4 - 576k^4p_x^4p_y^2 + 384k^4p_x^2p_y^4 - 64k^4p_y^6 + 256k^3p_x^5p_z^2 - 512k^3p_x^3p_y^2p_z^2 - \\
& 768k^3p_x^2p_y^4p_z^2 + 1152k^2p_x^6p_y^2 + 256k^2p_x^6p_z^2 + 384k^2p_x^4p_y^4 + 768k^2p_x^4p_y^2p_z^2 + 256k^2p_x^4p_z^4 - 640k^2p_x^2p_y^6 + \\
& 768k^2p_x^2p_y^4p_z^2 + 512k^2p_x^2p_y^2p_z^4 + 128k^2p_y^8 + 256k^2p_y^6p_z^2 + 256k^2p_y^4p_z^4 - 256k^2p_x^7p_z^2 + 256k^2p_x^5p_y^2p_z^2 + \\
& 1280k^2p_x^3p_y^4p_z^2 + 768k^2p_x^2p_y^6p_z^2 - 576k^2p_x^8p_y^2 - 256k^2p_x^8p_z^2 - 768k^2p_x^6p_y^4 - 1024k^2p_x^6p_y^2p_z^2 - 256k^2p_x^6p_z^4 + 128k^2p_x^4p_y^6 - \\
& 1536k^2p_x^4p_y^4p_z^2 - 768k^2p_x^4p_y^2p_z^4 + 256k^2p_x^8p_y^2 - 1024k^2p_x^2p_y^6p_z^2 - 768k^2p_x^2p_y^4p_z^4 - 64p_y^{10} - 256k^2p_y^8p_z^2 - 256k^2p_y^6p_z^4]
\end{aligned}$$

Parallel Singularities: Projection in (p_z, q_2, q_3) for OM^1

$$\begin{aligned}
& [s_3^2 - 3, 4k^7q_2^8 - 48k^7q_2^4q_3^4 - 64k^7q_2^2q_3^6 + 120k^7q_2^0q_3^8 + 384k^7q_2^8q_3^0 + 400k^7q_2^6q_3^2 + 192k^7q_2^4q_3^4 + \\
& 36k^7q_2^2q_3^6 + 8k^6q_2^6 - 8k^6q_2^4q_3^2 - 88k^6q_2^2q_3^4 - 40k^6q_2^0q_3^6 + 280k^6q_2^8q_3^0 + 488k^6q_2^6q_3^2 + 312k^6q_2^4q_3^4 + \\
& 72k^6q_2^2q_3^6 - 8k^7q_2^4 + 16k^7q_2^2q_3^2 + 72k^7q_2^0q_3^4 - 32k^7q_2^8q_3^0 - 248k^7q_2^6q_3^2 - 240k^7q_2^4q_3^4 - 72k^7q_2^2q_3^6 + \\
& 12k^5p_z^2q_2^4 - 132k^5p_z^2q_2^2q_3^2 + 828k^5p_z^2q_2^0q_3^4 - 756k^5p_z^2q_2^8q_3^0 - 540k^5p_z^2q_2^6q_3^2 + 1140k^5p_z^2q_2^4q_3^4 - \\
& 44k^5p_z^2q_2^2q_3^6 + 4k^5p_z^2q_3^4 - 16k^6q_2^4 + 32k^6q_2^2q_3^2 + 144k^6q_2^0q_3^4 - 64k^6q_2^8q_3^0 - 496k^6q_2^6q_3^2 - \\
& 480k^6q_2^4q_3^4 - 144k^6q_2^2q_3^6 - 8k^5p_z^2q_2^2 + 168k^5p_z^2q_2^0q_3^2 - 912k^5p_z^2q_2^8q_3^0 + 1808k^5p_z^2q_2^6q_3^2 - 1128k^5p_z^2q_2^4q_3^4 + \\
& 72k^5p_z^2q_2^2q_3^6 - 12k^5q_2^4 + 24k^5q_2^2q_3^2 + 108k^5q_2^0q_3^4 - 48k^5q_2^8q_3^0 - 372k^5q_2^6q_3^2 - 360k^5q_2^4q_3^4 - \\
& 108k^5q_2^2q_3^6 + 16k^4p_z^2q_2^2 + 96k^4p_z^2q_2^0q_3^2 + 240k^4p_z^2q_2^8q_3^0 + 320k^4p_z^2q_2^6q_3^2 + 240k^4p_z^2q_2^4q_3^4 + 96k^4p_z^2q_2^2q_3^6 + \\
& 16k^4p_z^2q_3^4 + 4k^7q_2^0 - 16k^7q_2^8q_3^0 - 8k^7q_2^6q_3^2 + 48k^7q_2^4q_3^4 + 36k^7q_2^2q_3^6 + k^5p_z^2q_2^0 - 67k^5p_z^2q_2^8q_3^0 - \\
& 86k^5p_z^2q_2^6q_3^2 + 26k^5p_z^2q_2^4q_3^4 + 37k^5p_z^2q_2^2q_3^6 - 7k^5p_z^2q_3^0 + 12k^3p_z^4q_2^0 - 120k^3p_z^4q_2^8q_3^0 - 120k^3p_z^4q_2^6q_3^2 + \\
& 160k^3p_z^4q_2^4q_3^4 + 140k^3p_z^4q_2^2q_3^6 - 8k^3p_z^4q_3^0 + 16k^6q_2^0 - 64k^6q_2^8q_3^0 - 32k^6q_2^6q_3^2 + 192k^6q_2^4q_3^4 + 144k^6q_2^2q_3^6 - \\
& 144k^4p_z^2q_2^8q_3^0 - 192k^4p_z^2q_2^6q_3^2 + 32k^4p_z^2q_2^4q_3^4 + 64k^4p_z^2q_2^2q_3^6 - 16k^4p_z^2q_3^0 - 16k^4q_2^2 + 48k^4q_2^0q_3^2 + \\
& 96k^4q_2^8q_3^0 - 160k^4q_2^6q_3^2 - 336k^4q_2^4q_3^4 - 144k^4q_2^2q_3^6 - 7k^5p_z^2q_2^8 + 44k^5p_z^2q_2^6q_3^2 - 18k^5p_z^2q_2^4q_3^4 - \\
& 68k^5p_z^2q_2^2q_3^6 + k^5p_z^2q_3^8 + 40k^5q_2^0 - 160k^5q_2^8q_3^0 - 80k^5q_2^6q_3^2 + 480k^5q_2^4q_3^4 + 360k^5q_2^2q_3^6 - 16k^3p_z^4q_2^8 + \\
& 152k^3p_z^4q_2^6q_3^2 - 24k^3p_z^4q_2^4q_3^4 - 184k^3p_z^4q_2^2q_3^6 + 8k^3p_z^4q_3^0 + 8k^3p_z^2q_2^0 - 32k^3p_z^2q_2^8q_3^0 - 16k^3p_z^2q_2^6q_3^2 + \\
& 96k^3p_z^2q_2^4q_3^4 + 72k^3p_z^2q_2^2q_3^6 - 8k^6q_2^8 + 40k^6q_2^6q_3^2 - 24k^6q_2^4q_3^4 - 72k^6q_2^2q_3^6 - 44k^4p_z^2q_2^8 + 112k^4p_z^2q_2^6q_3^2 - \\
& 168k^4p_z^2q_2^4q_3^4 - 336k^4p_z^2q_2^2q_3^6 - 12k^4p_z^2q_3^8 + 48k^4q_2^0 - 192k^4q_2^8q_3^0 - 96k^4q_2^6q_3^2 + 576k^4q_2^4q_3^4 + \\
& 432k^4q_2^2q_3^6 + 8k^2p_z^4q_2^8 - 184k^2p_z^4q_2^6q_3^2 - 24k^2p_z^4q_2^4q_3^4 + 152k^2p_z^4q_2^2q_3^6 - 16k^2p_z^4q_3^8 + 3k^5p_z^2q_2^6 + \\
& 9k^5p_z^2q_2^4q_3^4 + 9k^5p_z^2q_2^2q_3^6 + 3k^5p_z^2q_3^8 - 32k^5q_2^8 + 160k^5q_2^6q_3^2 - 96k^5q_2^4q_3^4 - 288k^5q_2^2q_3^6 + 14k^3p_z^4q_2^6 - \\
& 30k^3p_z^4q_2^4q_3^4 + 90k^3p_z^4q_2^2q_3^6 + 6k^3p_z^4q_3^8 - 56k^3p_z^2q_2^8 + 64k^3p_z^2q_2^6q_3^2 - 240k^3p_z^2q_2^4q_3^4 - 384k^3p_z^2q_2^2q_3^6 - \\
& 24k^3p_z^2q_3^8 + 16k^3q_2^0 - 64k^3q_2^8q_3^0 - 32k^3q_2^6q_3^2 + 192k^3q_2^4q_3^4 + 144k^3q_2^2q_3^6 + 4kp_z^6q_2^6 + 12kp_z^6q_2^4q_3^2 + \\
& 12kp_z^6q_2^2q_3^4 + 4kp_z^6q_3^6 + 32k^4p_z^2q_2^6 - 48k^4p_z^2q_2^4q_3^4 + 192k^4p_z^2q_2^2q_3^6 + 16k^4p_z^2q_3^8 - 48k^4q_2^8 + 240k^4q_2^6q_3^2 - \\
& 144k^4q_2^4q_3^4 - 432k^4q_2^2q_3^6 + 16k^2p_z^4q_2^6 + 192k^2p_z^4q_2^4q_3^4 - 48k^2p_z^4q_2^2q_3^6 + 32k^2p_z^4q_3^8 - 16k^2p_z^2q_2^8 - \\
& 208k^2p_z^2q_2^6q_3^2 - 144k^2p_z^2q_2^4q_3^4 + 16k^2p_z^2q_2^2q_3^6 - 32k^2p_z^2q_3^8 - k^5p_z^2q_2^4 - 2k^5p_z^2q_2^2q_3^6 - k^5p_z^2q_3^8 + 4k^5q_2^6 - \\
& 24k^5q_2^4q_3^4 + 36k^5q_2^2q_3^6 - 8k^3p_z^4q_2^4 - 16k^3p_z^4q_2^2q_3^6 - 8k^3p_z^4q_3^8 + 60k^3p_z^2q_2^6 - 36k^3p_z^2q_2^4q_3^4 + 324k^3p_z^2q_2^2q_3^6 + \\
& 36k^3p_z^2q_3^8 - 32k^3q_2^8 + 160k^3q_2^6q_3^2 - 96k^3q_2^4q_3^4 - 288k^3q_2^2q_3^6 - 8kp_z^6q_2^4 - 16kp_z^6q_2^2q_3^6 - 8kp_z^6q_3^8 + 20kp_z^4q_2^6 + \\
& 24kp_z^4q_2^4q_3^4 + 84kp_z^4q_2^2q_3^6 + 16kp_z^4q_3^8 - 4k^4p_z^2q_2^4 - 8k^4p_z^2q_2^2q_3^6 - 4k^4p_z^2q_3^8 + 16k^4q_2^6 - 96k^4q_2^4q_3^4 + \\
& 144k^4q_2^2q_3^6 - 20k^2p_z^4q_2^4 - 40k^2p_z^4q_2^2q_3^6 - 20k^2p_z^4q_3^8 + 32k^2p_z^2q_2^6 + 240k^2p_z^2q_2^4q_3^4 + 48k^2p_z^2q_3^8 + 2k^3p_z^4q_2^2 + \\
& 2k^3p_z^4q_3^8 - 12k^3p_z^2q_2^4 - 24k^3p_z^2q_2^2q_3^6 - 12k^3p_z^2q_3^8 + 16k^3q_2^6 - 96k^3q_2^4q_3^4 + 144k^3q_2^2q_3^6 + 5kp_z^6q_2^2 + \\
& 5kp_z^6q_3^8 - 24kp_z^4q_2^4 - 48kp_z^4q_2^2q_3^6 - 24kp_z^4q_3^8 + 16kp_z^2q_2^6 + 48kp_z^2q_2^4q_3^4 + 48kp_z^2q_2^2q_3^6 + 16kp_z^2q_3^8 +
\end{aligned}$$

$$4k^2p_z^4q_2^2 + 4k^2p_z^4q_3^2 - 16k^2p_z^2q_2^4 - 32k^2p_z^2q_2^2q_3^2 - 16k^2p_z^2q_3^4 - kp_z^6 + 8kp_z^4q_2^2 + 8kp_z^4q_3^2 - 16kp_z^2q_2^4 - 32kp_z^2q_2^2q_3^2 - 16kp_z^2q_3^4]$$

Parallel Singularities: Projection in (p_z, q_2, q_3) for OM^2

$$\begin{aligned} & [k(36k^6q_2^1q_3^2 + 192k^6q_2^1q_3^4 + 400k^6q_2^1q_3^6 + 384k^6q_2^1q_3^8 + 120k^6q_2^8q_3^0 - 64k^6q_2^6q_3^2 - 48k^6q_2^4q_3^4 + \\ & 4k^6q_3^8 - 72k^5q_2^1q_3^2 - 312k^5q_2^1q_3^4 - 488k^5q_2^1q_3^6 - 280k^5q_2^8q_3^0 + 40k^5q_2^6q_3^2 + 88k^5q_2^4q_3^4 + \\ & 8k^5q_2^2q_3^6 - 8k^5q_3^8 - 72k^6q_2^1q_3^2 - 240k^6q_2^1q_3^4 - 248k^6q_2^8q_3^0 - 32k^6q_2^6q_3^2 + 72k^6q_2^4q_3^4 + 16k^6q_2^2q_3^6 - \\ & 8k^6q_3^8 + 4k^4p_z^2q_2^1q_3^2 - 44k^4p_z^2q_2^1q_3^4 + 1140k^4p_z^2q_2^1q_3^6 - 540k^4p_z^2q_2^8q_3^0 - 756k^4p_z^2q_2^6q_3^2 + 828k^4p_z^2q_2^4q_3^4 - \\ & 132k^4p_z^2q_2^2q_3^6 + 12k^4p_z^2q_3^8 + 144k^5q_2^1q_3^2 + 480k^5q_2^1q_3^4 + 496k^5q_2^8q_3^0 + 64k^5q_2^6q_3^2 - 144k^5q_2^4q_3^4 - \\ & 32k^5q_2^2q_3^6 + 16k^5q_3^8 + 72k^4p_z^2q_2^1q_3^2 - 1128k^4p_z^2q_2^1q_3^4 + 1808k^4p_z^2q_2^1q_3^6 - 912k^4p_z^2q_2^8q_3^0 + 168k^4p_z^2q_2^6q_3^2 - \\ & 8k^4p_z^2q_2^4q_3^4 - 108k^4q_2^1q_3^2 - 360k^4q_2^1q_3^4 - 372k^4q_2^8q_3^0 - 48k^4q_2^6q_3^2 + 108k^4q_2^4q_3^4 + 24k^4q_2^2q_3^6 - \\ & 12k^4q_3^8 - 16k^3p_z^2q_2^1q_3^2 - 96k^3p_z^2q_2^1q_3^4 - 240k^3p_z^2q_2^8q_3^0 - 320k^3p_z^2q_2^6q_3^2 - 240k^3p_z^2q_2^4q_3^4 - 96k^3p_z^2q_2^2q_3^6 - \\ & 16k^3p_z^2q_3^8 + 36k^6q_2^6q_3^2 + 48k^6q_2^4q_3^4 - 8k^6q_2^2q_3^6 - 16k^6q_2^0q_3^8 + 4k^6q_3^0 - 7k^4p_z^2q_2^1q_3^2 + 37k^4p_z^2q_2^8q_3^0 + \\ & 26k^4p_z^2q_2^6q_3^2 - 86k^4p_z^2q_2^4q_3^4 - 67k^4p_z^2q_2^2q_3^6 + k^4p_z^2q_3^8 - 8k^2p_z^4q_2^1q_3^2 + 140k^2p_z^4q_2^8q_3^0 + 160k^2p_z^4q_2^6q_3^2 - \\ & 120k^2p_z^4q_2^4q_3^4 - 120k^2p_z^4q_2^2q_3^6 + 12k^2p_z^4q_3^8 - 144k^5q_2^8q_3^0 - 192k^5q_2^6q_3^2 + 32k^5q_2^4q_3^4 + 64k^5q_2^2q_3^6 - \\ & 16k^5q_3^8 + 16k^3p_z^2q_2^1q_3^2 - 64k^3p_z^2q_2^8q_3^0 - 32k^3p_z^2q_2^6q_3^2 + 192k^3p_z^2q_2^4q_3^4 + 144k^3p_z^2q_2^2q_3^6 + 144k^3q_2^1q_3^2 + \\ & 336k^3q_2^8q_3^0 + 160k^3q_2^6q_3^2 - 96k^3q_2^4q_3^4 - 48k^3q_2^2q_3^6 + 16k^3q_3^8 + k^4p_z^2q_2^8 - 68k^4p_z^2q_2^6q_3^2 - 18k^4p_z^2q_2^4q_3^4 + \\ & 44k^4p_z^2q_2^2q_3^6 - 7k^4p_z^2q_3^8 + 360k^4q_2^8q_3^0 + 480k^4q_2^6q_3^2 - 80k^4q_2^4q_3^4 - 160k^4q_2^2q_3^6 + 40k^4q_3^8 + 8k^2p_z^4q_2^8 - \\ & 184k^2p_z^4q_2^6q_3^2 - 24k^2p_z^4q_2^4q_3^4 + 152k^2p_z^4q_2^2q_3^6 - 16k^2p_z^4q_3^8 + 72k^2p_z^2q_2^8q_3^0 + 96k^2p_z^2q_2^6q_3^2 - 16k^2p_z^2q_2^4q_3^4 - \\ & 32k^2p_z^2q_2^2q_3^6 + 8k^2p_z^2q_3^8 + 72k^5q_2^6q_3^2 + 24k^5q_2^4q_3^4 - 40k^5q_2^2q_3^6 + 8k^5q_3^8 + 12k^3p_z^2q_2^8 + 336k^3p_z^2q_2^6q_3^2 + \\ & 168k^3p_z^2q_2^4q_3^4 - 112k^3p_z^2q_2^2q_3^6 + 44k^3p_z^2q_3^8 - 432k^3q_2^8q_3^0 - 576k^3q_2^6q_3^2 + 96k^3q_2^4q_3^4 + 192k^3q_2^2q_3^6 - \\ & 48k^3q_3^8 + 16kp_z^4q_2^8 - 152kp_z^4q_2^6q_3^2 + 24kp_z^4q_2^4q_3^4 + 184kp_z^4q_2^2q_3^6 - 8kp_z^4q_3^8 + 3k^4p_z^2q_2^6 + 9k^4p_z^2q_2^4q_3^2 + \\ & 9k^4p_z^2q_2^2q_3^4 + 3k^4p_z^2q_3^6 - 288k^4q_2^6q_3^2 - 96k^4q_2^4q_3^4 + 160k^4q_2^2q_3^6 - 32k^4q_3^8 + 6k^2p_z^4q_2^6 + 90k^2p_z^4q_2^4q_3^2 - \\ & 30k^2p_z^4q_2^2q_3^4 + 14k^2p_z^4q_3^6 - 24k^2p_z^2q_2^8 - 384k^2p_z^2q_2^6q_3^2 - 240k^2p_z^2q_2^4q_3^4 + 64k^2p_z^2q_2^2q_3^6 - 56k^2p_z^2q_3^8 + \\ & 144k^2q_2^8q_3^0 + 192k^2q_2^6q_3^2 - 32k^2q_2^4q_3^4 - 64k^2q_2^2q_3^6 + 16k^2q_3^8 + 4p_z^6q_2^6 + 12p_z^6q_2^4q_3^2 + 12p_z^6q_2^2q_3^4 + \\ & 4p_z^6q_3^6 - 16k^3p_z^2q_2^6 - 192k^3p_z^2q_2^4q_3^2 + 48k^3p_z^2q_2^2q_3^4 - 32k^3p_z^2q_3^6 + 432k^3q_2^6q_3^2 + 144k^3q_2^4q_3^4 - 240k^3q_2^2q_3^6 + \\ & 48k^3q_3^8 - 32kp_z^4q_2^6 + 48kp_z^4q_2^4q_3^2 - 192kp_z^4q_2^2q_3^4 - 16kp_z^4q_3^6 + 32kp_z^2q_2^8 - 16kp_z^2q_2^6q_3^2 + 144kp_z^2q_2^4q_3^4 + \\ & 208kp_z^2q_2^2q_3^6 + 16kp_z^2q_3^8 - k^4p_z^2q_2^4 - 2k^4p_z^2q_2^2q_3^4 - k^4p_z^2q_3^6 + 36k^4q_2^4q_3^2 - 24k^4q_2^2q_3^4 + 4k^4q_3^6 - 8k^2p_z^4q_2^4 - \\ & 16k^2p_z^4q_2^2q_3^2 - 8k^2p_z^4q_3^4 + 36k^2p_z^2q_2^6 + 324k^2p_z^2q_2^4q_3^2 - 36k^2p_z^2q_2^2q_3^4 + 60k^2p_z^2q_3^6 - 288k^2q_2^6q_3^2 - \\ & 96k^2q_2^4q_3^4 + 160k^2q_2^2q_3^6 - 32k^2q_3^8 - 8p_z^6q_2^4 - 16p_z^6q_2^2q_3^2 - 8p_z^6q_3^4 + 16p_z^4q_2^6 + 84p_z^4q_2^4q_3^2 + 24p_z^4q_2^2q_3^4 + \\ & 20p_z^4q_3^6 + 4k^3p_z^2q_2^4 + 8k^3p_z^2q_2^2q_3^2 + 4k^3p_z^2q_3^4 - 144k^3q_2^4q_3^2 + 96k^3q_2^2q_3^4 - 16k^3q_3^6 + 20kp_z^4q_2^4 + 40kp_z^4q_2^2q_3^2 + \\ & 20kp_z^4q_3^4 - 48kp_z^2q_2^6 - 240kp_z^2q_2^4q_3^2 - 32kp_z^2q_3^6 + 2k^2p_z^4q_2^2 + 2k^2p_z^4q_3^2 - 12k^2p_z^2q_2^4 - 24k^2p_z^2q_2^2q_3^2 - \\ & 12k^2p_z^2q_3^4 + 144k^2q_2^4q_3^2 - 96k^2q_2^2q_3^4 + 16k^2q_3^6 + 5p_z^6q_2^2 + 5p_z^6q_3^2 - 24p_z^4q_2^4 - 48p_z^4q_2^2q_3^2 - 24p_z^4q_3^4 + \\ & 16p_z^2q_2^6 + 48p_z^2q_2^4q_3^2 + 48p_z^2q_2^2q_3^4 + 16p_z^2q_3^6 - 4kp_z^4q_2^2 - 4kp_z^4q_3^2 + 16kp_z^2q_2^4 + 32kp_z^2q_2^2q_3^2 + 16kp_z^2q_3^4 - \\ & p_z^6 + 8p_z^4q_2^2 + 8p_z^4q_3^2 - 16p_z^2q_2^4 - 32p_z^2q_2^2q_3^2 - 16p_z^2q_3^4)] \end{aligned}$$

B

SIROPA Library

B.1 SIROPA

Analysing

ConstraintEquations	Computes the implicit equations induced by the constraints.
Type1SingularityEquations / SerialSingularities	Computes the implicit equations satisfied by the singularities.
Type2SingularityEquations / ParallelSingularities	Computes the implicit equations satisfied by the singularities.
InfiniteEquations	Computes the equations where the manipulator has infinitely many solutions.
ParallelCuspidal	Computes the implicit equations satisfied by the cuspidal points
SerialCuspidal	Computes the implicit equations satisfied by the cuspidal points
Projection	Project on variables or expressions in a polynomial system.
CellDecompositionPlus	Describes the parameter space according to the number of real roots.
NumberOfSolutionsPlus	Returns the number of real solutions of all cells obtained by CellDecompositionPlus .
auxDVNumberOfSolutionsPlus	Auxiliary function Returns the number of real solutions on the intersection points of the Discriminant Variety.
PseudoSingularitiesDecomposition	Computes the decomposition with pseudo-singularities of a manipulator.
IsIntervalEmpty	Checks that a polynomial has no real roots in an open interval.
CellGraph	Computes the connexity graph of the cells of a CAD.
ImageTrajectory	Display a given trajectory.
DirectInverseKinematics	Returns the cells having the same antecedent.
CellArea2D	Compute the area of the cells returned CellDecomposition or CellDecompositionPlus .
CellLocationPlus	Computes the cells of a decomposition containing the given points.

Table B.1 – Different Functions for Analysing in SIROPA

Modeling

CreateManipulator	Constructs a data structure of type Manipulator.
SubsPlus	Substitute coherently angles in a system.
SubsParameters	Specify parameter values in a Manipulator
UnassignParameters	Release parameters in a Manipulator

Table B.2 – Different Functions for Modeling in SIROPA

Plotting

Plot2D	Plots a system of 2 variables
PlotCurve3D	Plots a curve given by implicit equations in 3 variables.
Plot3D	Plots a system of 3 variables using maple internal plotting functions.
Plot3Dglsurf	Plots a system of 3 variables using glsurf.
Plot3Dsurfex	Plots a system of 3 variables using surfex (software based on surf).
PlotWorkspace	Plot the border of a manipulator workspace
Configurations	Computes the different possible positions.
PlotRobot2D	Plot a planar manipulator
PlotRobot3D	Plot a 3D manipulator
PlotCell3D	Plot the cells returned CellDecomposition or CellDecompositionPlus
PlotCell2D	Plot the cells returned CellDecomposition or CellDecompositionPlus
SetCellColors	Set colors to the numbers of solutions obtained by NumberOfSolutionsPlus
Trajectory	Display a given trajectory.
ImageTrajectory	Display a given trajectory.

Table B.3 – Different Functions for Plotting in SIROPA

Plot2D

```

1 Plot2D := proc (
2   sys          :: { algebraic , equation(algebraic) , list({ algebraic , equation(
   algebraic) , algebraic < algebraic } ) , list( list({ algebraic , equation(algebraic) ,
   algebraic < algebraic } ) ) } ,
3   e1           :: name                = range ,
4   e2           :: name                = range ,
5   {
6   points       :: truefalse           := false ,
7   [ notest , draft ] :: truefalse     := false
8   }
9 )

```

Listing B.1 – Architecture of Plot2D

Plots a system of 2 variables

Parameters

<code>sys</code>	a list or a list of list of polynomials: the system
<code>v1 = r1</code>	<code>v1</code> is a name of <code>sys</code> and <code>r1</code> a range of values
<code>v2 = r2</code>	<code>v2</code> is a name of <code>sys</code> and <code>r2</code> a range of values
<code>points = bool (optional)</code>	<code>bool</code> is a boolean: if false, isolated points are ignored; default value: false;
<code>notest = b (optional)</code>	<code>b</code> is a boolean; when <code>b</code> is true the inequality and real constraints are ignored; default value: true;
<code>opts (optional)</code>	arguments passed to the maple function <code>plots:-implicitplot</code>

Returns

A graphic: the solutions of the system.

- when `sys` is a polynomial, the graphic is the zeroes of this polynomial
- when `sys` is a list of polynomials $[p_1, \dots, p_k]$, the graphic is the zeroes of the system $p_1 = 0$ and ... and $p_k = 0$
- when `sys` is a list of list of polynomials $[L_1, \dots, L_k]$, the graphic is the union of the zeroes of each system L_1, \dots, L_k .

PlotCurve3D

```

1 PlotCurve3D := proc (
2   sys      :: { algebraic , equation(algebraic) , list({ algebraic , equation(algebraic)
3     , algebraic < algebraic } ) , list ( list ( { algebraic , equation(algebraic) , algebraic <
4     algebraic } ) ) } ,
5   ineq     :: list ( { polynomial , polynomial < polynomial } ) := [] ,
6   e1       :: name
7             = range := sort
8             ([ op ( indets ( [ sys , x , y , z ] , name ) ) )
9             [ 1 ] = -5..5 ,
10            = range := sort
11            ([ op ( indets ( [ sys , x , y , z ] , name )
12              minus { lhs ( e1 ) } ) )
13            [ 1 ] = -5..5 ,
14            = range := sort
15            ([ op ( indets ( [ sys , x , y , z ] , name )
16              minus { lhs ( e1 ) , lhs ( e2 ) } ) )
17            [ 1 ] = -5..5 ,
18            {
19   grid      :: integer          := 30 ,
20   output    :: identical ( list , display ) := ':-display '
21   }
22 )

```

Listing B.2 – Architecture of PlotCurve3D

Plots a curve given by implicit equations in 3 variables.

Parameters

<code>sys</code>	a list or a list of list of polynomials: the system
<code>v1 = r1 v1</code>	is a name of <code>sys</code> and <code>r1</code> a range of values
<code>v2 = r2 v2</code>	is a name of <code>sys</code> and <code>r2</code> a range of values
<code>v3 = r3 v3</code>	is a name of <code>sys</code> and <code>r3</code> a range of values
<code>grid = i (optional)</code>	<code>i</code> is an integer, the number of samples points; default value: 20.
<code>output = keyword (optional)</code>	keyword is either <code>list</code> or <code>display</code> : <code>display</code> (resp. <code>list</code>) returns a graph (resp. a list).

Returns

A graphic: the solutions of the system.

- when `sys` is a list of polynomials $[p_1, \dots, p_k]$, the graphic is the zeroes of the system $p_1 = 0$ and ... and $p_k = 0$
- when `sys` is a list of list of polynomials $[L_1, \dots, L_k]$, the graphic is the union of the zeroes of each system L_1, \dots, L_k .

Plot3D

```

1 Plot3D := proc (
2   sys          ::{ algebraic , equation(algebraic) , list({ algebraic , equation(
3     algebraic) , algebraic<algebraic} ) , list(list({ algebraic , equation(algebraic) ,
4     algebraic<algebraic} ))} ,
5   ineq         ::list({polynom , polynom<polynom}) := [] ,
6   e1           ::name          = range := sort
7                                     ([op(indets([sys , x , y , z] , name))]
8                                     [1] = -5..5 ,
9   e2           ::name          = range := sort
10                                    ([op(indets([sys , x , y , z] , name)
11                                    minus {lhs(e1)}))]
12                                    [1] = -5..5 ,
13   e3           ::name          = range := sort
14                                    ([op(indets([sys , x , y , z] , name)
15                                    minus {lhs(e1) , lhs(e2)}))]
16                                    [1] = -5..5 ,
17   {
18     points          ::truefalse          := false ,
19     crossingrefine  ::truefalse          := false ,
20     grid            ::integer            := 10 ,
21     border          ::constant           := 10(-30) ,
22     output          ::identical(list , display) := ':-display'
23   }
24 )

```

Listing B.3 – Architecture of Plot3D

Plots a system of 3 variables using maple internal plotting functions.

Parameters

<code>sys</code>	a list or a list of list of polynomials: the system
<code>v1 = r1 v1</code>	is a name of <code>sys</code> and <code>r1</code> a range of values
<code>v2 = r2 v2</code>	is a name of <code>sys</code> and <code>r2</code> a range of values
<code>v3 = r3 v3</code>	is a name of <code>sys</code> and <code>r3</code> a range of values
<code>points = bool (optional)</code>	<code>bool</code> is a boolean: if false, isolated points are ignored; default value: false;
<code>grid = i (optional)</code>	<code>i</code> is an integer leading to a grid size $i \times i$; default value: 20.
<code>border = e (optional)</code>	<code>e</code> is a numeric value: defines the precision on the border; default value: 0.0001.
<code>crossingrefine = bool (optional)</code>	<code>bool</code> is a boolean: if true, the mesh follows the cross of the different surfaces; default value: false.
<code>output = keyword (optional)</code>	<code>keyword</code> is either list or display: display (resp. list) returns a graph (resp. a list).

Returns

A graphic: the solutions of the system.

- when `sys` is a polynomial, the graphic is the zeroes of this polynomial
- when `sys` is a list of polynomials $[p1, \dots, pk]$, the graphic is the zeroes of the system $p1 = 0$ and ... and $pk = 0$
- when `sys` is a list of list of polynomials $[L1, \dots, Lk]$, the graphic is the union of the zeroes of each system $L1, \dots, Lk$.

Plot3Dglsurf

```

1 Plot3Dglsurf := proc (
2   sys      ::{ algebraic , equation(algebraic) , list({ algebraic , equation(algebraic)
3     } , list(list({ algebraic , equation(algebraic)}))} ,
4   e1       ::name           = range ,
5   e2       ::name           = range ,
6   e3       ::name           = range ,
7   {
8     points  ::truefalse     := false ,
9     scale   ::identical(constrained , unconstrained) := 'unconstrained'
10  }
11 )

```

Listing B.4 – Architecture of Plot3Dglsurf

Plots a system of 3 variables using `glsurf`.

Parameters

<code>sys</code>	a list or a list of list of polynomials: the system
<code>v1 = r1</code>	<code>v1</code> is a name of <code>sys</code> and <code>r1</code> a range of values
<code>v2 = r2</code>	<code>v2</code> is a name of <code>sys</code> and <code>r2</code> a range of values
<code>v3 = r3</code>	<code>v3</code> is a name of <code>sys</code> and <code>r3</code> a range of values
<code>points = bool (optional)</code>	<code>bool</code> is a boolean: if false, isolated points are ignored; default value: false;
<code>scaled = keyword (optional)</code>	<code>keyword</code> is one of constrained or unconstrained for the scaling; default value: <code>scaled = unconstrained</code> .

Returns

A graphic: the solutions of the system.

- when `sys` is a polynomial, the graphic is the zeroes of this polynomial
- when `sys` is a list of polynomials $[p_1, \dots, p_k]$, the graphic is the zeroes of the system $p_1 = 0$ and ... and $p_k = 0$
- when `sys` is a list of list of polynomials $[L_1, \dots, L_k]$, the graphic is the union of the zeroes of each system L_1, \dots, L_k .

PlotWorkspace

```

1 PlotWorkspace := proc (
2   robot      :: Manipulator ,
3   spec       :: seq (name=constant) ,
4   disp       :: seq (identical (type1, type2, constraints, infinite)) := ('type1',
5                                     'type2', 'constraints', 'infinite'),
6   vars       :: { list (name), identical (articular, pose) }           := 'articular',
7   ranges     :: seq (name=range)                                       := NULL,
8   {
9   notest     :: truefalse                                             := false
10  }
11 )
12 )

```

Listing B.5 – Architecture of PlotWorkspace

Plot the border of a manipulator workspace

Parameters

robot	a object of type Manipulator
spec	sequence of name=constant: the specification of the known variables (the articular values or the pose values, or other)
disp (optional)	a sequence of keywords among type1, type2, constraints and infinite: selects the graphics to plot; default value: all the keywords the display variables, that can be referred as: one of the keyword articular or pose
vars (optional)	a list of 2 variable names the default value is articular if ranges is provided, the variables superseed those of vars
ranges (optional)	a sequence of the form v1=n1..n2, v2=m1..m2 where v1, v2 are 2 variables and n1,n2,m1,m2 are numerical values; when ranges is not specified, the default range is taken from the field robot:-DefaultPlotRange if it exists, and is -5..5 otherwise.
notest = b (optional)	b is a boolean; when b is true the inequality and real constraints are ignored; default value: true;

Returns

A graphic: the solutions of the system.

- The singularities of type 1
- The singularities of type 2
- The configurations with infinitely many solutions in all the variables (including the passive variables)
- The border induced by the constraints

PlotWorkspace

```

1 PlotWorkspace := proc (
2   robot      :: Manipulator ,
3   spec       :: seq(name=constant) ,
4   disp       :: seq(identical(type1,type2, constraints,infinite)) := ('type1',
5                                     'type2', 'constraints', 'infinite'),
6   vars       :: { list(name), identical(articular, pose) }         := 'articular',
7   ranges     :: seq(name=range)                                   := NULL,
8   {
9   notest     :: truefalse                                         := false
10  }
11 )
12 )

```

Listing B.6 – Architecture of PlotWorkspace

Plot the border of a manipulator workspace

Parameters

robot	a object of type Manipulator
spec	sequence of name=constant: the specification of the known variables (the articular values or the pose values, or other)
disp (optional)	a sequence of keywords among type1, type2, constraints and infinite: selects the graphics to plot; default value: all the keywords the display variables, that can be referred as: one of the keyword articular or pose
vars (optional)	a list of 2 variable names the default value is articular if ranges is provided, the variables supersede those of vars
ranges (optional)	a sequence of the form $v1=n1..n2$, $v2=m1..m2$ where $v1$, $v2$ are 2 variables and $n1, n2, m1, m2$ are numerical values; when ranges is not specified, the default range is taken from the field robot:-DefaultPlotRange if it exists, and is -5..5 otherwise.
notest = b (optional)	b is a boolean; when b is true the inequality and real constraints are ignored; default value: true;

Returns

A graphic: the solutions of the system.

- The singularities of type 1
- The singularities of type 2
- The configurations with infinitely many solutions in all the variables (including the passive variables)
- The border induced by the constraints

B.2 Algebraic Tools

Base Algebra

Projection	Project on variables or expressions in a polynomial system.
TrigonometricBasis	Computes a Groebner basis of a trigonometric system.
Elimination	Eliminates variables in a polynomial system.
Intersection	Intersects two ideals.
Division	Division of an ideal by another.
NormalForm	Reduces polynomials w.r.t a Groebner basis.
InRadical	Test the membership to the radical of an ideal.
Dimension	Computes the dimension of a system.
SaturationIdeal	Saturate an ideal by another ideal.
SaturationTest	Saturate an ideal by another ideal (new method).
Saturation5	Saturate an ideal by another ideal (new method).
SaturationDirect	Saturate an ideal by another ideal (direct method).

Table B.4 – Different Functions for Base Algebra

Extra Algebra

PolynomialSaturation	Saturates an ideal by a polynomial.
SizeReduction	Reduces the number of polynomials.
IsHullPrime	Test whether the maximal dimension component is prime.
IsHullRadical	Test whether the maximal dimension component is radical.
IsProjectionProper	Sufficient condition for proper projection.
PropernessDefect	Computes the non properness set of a projection.
SortPolynomials	Sort a list of polynomial and trigonometric expressions
FactorSystem	Factorize partially a system
Hypersurfaces	Factorizes components of codimension 1.
RandomSection	Cuts a system with a random hypersurface
ToLatex	Print the latex commands generating an expression on multiple lines.

Table B.5 – Different Functions for Extra Algebra

Trigonometric

TrigonometricSubs	Substitute trigonometric variables with numerical values
TrigonometricAlgebraic	Convert trigonometric and radical expressions to algebraic ones
TrigonometricAlgebraicCosSin	Convert trigonometric and radical expressions to algebraic ones
TrigonometricVariables	Updates trigonometric variables to algebraic ones.
TrigonometricConstants	Returns the algebraic variables of trigonometric constants.
AlgebraicVariables	Convert algebraic variables to trigonometric ones.
AlgebraicTrigonometric	Convert algebraic systems back to trigonometric ones.
TrigonometricTanHalf	Parametrize trigonometric and radical expressions with tangent expression
TrigonometricGcd	Extract common factors of trigonometric expressions

Table B.6 – Different Functions for Trigonometric

C

Timeline

First Year of Thesis Timeline

Second Year of Thesis Timeline

Third Year of Thesis Timeline

Bibliography

- [1] Soni A. and Tsai Y.: An algorithm for the workspace of a general n-R robot. *ASME J. Mech. Trans. Automat. Des.*, Vol. 105, pp. 52–57 (1983). 3, 31
- [2] Hansen J. and Gupta K. C. and Kazerounian S. M. K.: Generation and evaluation of the workspace of a manipulator. *Int. J. Robot. Res.*, Vol. 2/3, pp. 22–31 (1983). 3, 31
- [3] Gosselin C.: Determination of the workspace of 6-DOF parallel manipulators. *ASME J. Mech. Des.*, vol. 112, pp. 331–336 (1990). 4, 31, 71
- [4] Merlet J. P.: Geometrical determination of the workspace of a constrained parallel manipulator. In: *Proceedings of 3rd Int. Symp. on Advances in Robot Kinematics*, pp. 326–329 (1992). 4, 31, 71
- [5] Castelli G., Ottaviano E. and Ceccarelli M.: A fairly general algorithm to evaluate workspace characteristics of serial and parallel manipulators. *Mech. Based Des. Struct. Mach.*, vol. 36/1, pp. 14–33 (2008). 4, 31, 71
- [6] Gogu G., 2005. Mobility of mechanisms: a critical review. *Mechanism and Machine Theory*, 40(9), pp.1068-1097. 11
- [7] Bonev I. A. and Ryu J.: A new approach to orientation workspace analysis of 6-DOF parallel manipulators. *Mechanism and Machine Theory*, Vol 36/1, pp. 15–28 (2001). 4, 15, 31, 32, 71
- [8] Chablat D., Wenger P., Majou F. and Merlet J.P.: An Interval Analysis Based Study for the Design and the Comparison of 3-DOF Parallel Kinematic Machines. *International Journal of Robotics Research*, Vol. 23/6, pp. 615–624 (2004). 4, 14, 31, 32, 71

- [9] Zein M., Wenger P. and Chablat D.: An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications. *Mechanism and Machine Theory*, Vol. 41/8, pp. 971–986 (2006). 4, 31, 71
- [10] Ottaviano E., Husty M. and Ceccarelli M.: Identification of the workspace boundary of a general 3-R manipulator. *ASME J. Mech. Des.*, Vol. 128/1, pp. 236–242 (2006). 4, 31, 71
- [11] Chablat D., Jha R. and Rouillier F. and Moroz G.: Non-singular assembly mode changing trajectories in the workspace for the 3-RPS parallel robot. 14th International Symposium on Advances in Robot Kinematics, Ljubljana, Slovenia, pp. 149–159 (2014). 4, 12, 31, 32, 71
- [12] Chablat D., Moroz G. and Wenger P.: Uniqueness domains and non singular assembly mode changing trajectories. In: *Proc. IEEE Int. Conf. Rob. and Automation* (2011). 4, 5, 20, 21, 22, 26, 31, 43, 44, 61, 62, 67, 71
- [13] Moroz G., Rouillier F., Chablat D., Wenger P., On the determination of cusp points of 3-RPR parallel manipulators, *Mechanism and Machine Theory*, Vol. 45 (11), (2010). 44, 68
- [14] Caro S., Wenger P., Chablat D., Non-Singular Assembly Mode Changing Trajectories of a 6-DOF Parallel Robot, *Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2012*, Chicago, Illinois, USA, August 12-15, (2012). 4, 25, 44, 52
- [15] Chablat D., Moroz G., Arakelian V., Briot S., Wenger P., Solution regions in the parameter space of a 3-RRR decoupled robot for a prescribed workspace, 13th International Symposium on Advances in Robot Kinematics, Kluwer Academic Publishers, Austria, June 24-28, (2012). 4
- [16] Chablat D., Jha R., Rouillier F. and Moroz G.: Workspace and joint space analysis of the 3-RPS parallel robot. In: *Proceedings of ASME 2014 International Design Engineering Technical Conferences*, Buffalo (2014). 4, 32, 39, 74
- [17] Wenger Ph., Chablat D., Definition Sets for the Direct Kinematics of Parallel Manipulators, 8th International Conference in Advanced Robotics, pp. 859–864, (1997). 4, 21, 41, 51, 61, 62, 63
- [18] Wenger P. and Chablat D., Workspace and assembly modes in fully parallel manipulators: a descriptive study, In: *ARK, Strobol*, pp. 117-126, (1998). 4, 25
- [19] Siciliano B. and Khatib O.: *Springer handbook of robotics*. Springer (2008). 4, 14, 31, 72
- [20] Snyman J. A., Plessis L. J. and Duffy J.: An optimization approach to the determination of the boundaries of manipulator workspaces. *ASME J. Mech. Des.*, Vol. 122/4, pp. 447–456 (2000). 4, 32
- [21] Bonev I. A. and Gosselin C. M.: Analytical determination of the workspace of symmetrical spherical parallel mechanisms. *IEEE Trans. Robot.*, Vol. 22/5, pp. 1011–1017 (2006). 4, 32

-
- [22] Corvez S, Rouillier F. Using computer algebra tools to classify serial manipulators. Springer Berlin Heidelberg; (2002). 4, 31
- [23] Faugere J., Rolland L., and Rouillier F., Using computer algebra tools for off-line studies of parallel manipulators, in International conference on applications of computer algebra , (1999). 4, 31
- [24] Rouillier F., Real roots counting for some robotics problems, in Computational kinematics, Kluwer, pp. 73-82, (1995). 4, 31
- [25] Merlet J.P., A generic trajectory verifier for the motion planning of parallel robots, Journal of mechanical design, Vol. 123(4), pp. 510-515, (2001). 5, 71
- [26] Dasgupta B., Mruthyunjaya T. S., Singularity-free path planning for the Stewart platform manipulator, Mechanism and Machine Theory, Vol. 33(6), pp. 711-725, (1998). 5, 71
- [27] Dash A. K., Chen I. M., Yeo S. H., Yang G., Singularity-free path planning of parallel manipulators using clustering algorithm and line geometry, In Robotics and Automation, 2003. Proceedings. ICRA'03, Vol. 1, pp. 761-766, (2003). 5, 71
- [28] Bhattacharya S., Hatwal H., Ghosh A., Comparison of an exact and an approximate method of singularity avoidance in platform type parallel manipulators, Mechanism and Machine Theory, Vol. 33(7), pp. 965-974, (1998). 5, 71
- [29] Rolland L., Certified solving of the forward kinematics problem with an exact algebraic method for the general parallel manipulator. Advanced Robotics, Vol. 19(9), pp. 995-1025, (2005). 72
- [30] Lahouar S., Zegloul S., Romdhane L., Singularity Free Path Planning for Parallel Robots, Jadran Lenarčič and Philippe Wenger (eds.), Advances in Robot Kinematics: Analysis and Design, pp. 235-242, (2008). 5, 72
- [31] Bohigas O., Henderson M. E., Ros L., Manubens M., Porta J. M., Planning singularity-free paths on closed-chain manipulators, Robotics, IEEE Transactions on, Vol. 29(4), pp. 888-898, (2013). 5, 72
- [32] Drouet P., Dubowsky S., Zegloul S., Mavroidis C., Compensation of geometric and elastic errors in large manipulators with an application to a high accuracy medical system. Robotica, Vol. 20(03), pp. 341-352, (2002). 5, 97
- [33] Niaritsiry T. F., Fazenda N., Clavel R., Study of the sources of inaccuracy of a 3 DOF flexure hinge-based parallel manipulator, In Robotics And Automation, 2004. Proceedings. Icara'04. 2004 IEEE International Conference, Vol. 4, pp. 4091-4096, (2004). 5, 97
- [34] Wang J. and Masory O., On the accuracy of a Stewart platform. I. The effect of manufacturing tolerances, in Robotics and Automation, Proceedings., 1993 IEEE International Conference on , vol. 1, pp. 114-120, (1993). 5, 97
- [35] Kim H. S., and Choi Y. J., The kinematic error bound analysis of the Stewart platform, Journal of Robotic Systems, Vol. 17(01), pp. 63-73, (2000). 5, 97

- [36] Caro S., Wenger P., Bennis F., Chablat D., Sensitivity Analysis of the Orthoglide: A Three-DOF Translational Parallel Kinematic Machine, *ASME. J. Mech. Des.*, Vol. 128(2), pp. 392–402, (2005). 6, 97
- [37] Pashkevich A., Klimchick A., Chablat D., Enhanced stiffness modeling of manipulators with passive joints, *Mechanism and Machine Theory*, Vol. 46(5), pp. 662–679, (2011). 6, 98
- [38] Pashkevich A., Chablat D., Wenger P., Kinematic calibration of Orthoglide-type mechanisms from observation of parallel leg motions, *Mechatronics*, Vol. 19(4), pp. 478–488, (2009). 6, 98, 100
- [39] Aginaga J., Altuzarra O., Macho E., Iriarte X., Assessing Position Error Due to Clearances and Deformations of Links in Parallel Manipulators, *ASME. J. Mech. Des.*, vol. 135(04), pp. 041006, (2013). 98
- [40] Meng J., Zhang D., Li Z., Accuracy Analysis of Parallel Manipulators With Joint Clearance, *ASME. J. Mech. Des.*, vol. 131(01), pp. 011013, (2008). 98
- [41] Rolland L., Methodes algebriques pour la resolution du modÁlle geometrique de robots paralleles, applications a haute cadence et grande precision, PhD Thesis, Université Henry Poincaré – Nancy, (2002). 6, 98
- [42] Vivas A., Poignet P., Predictive functional control of a parallel robot, *Control Eng. Pract.* 13(7), 863–874, (2005). 6, 98
- [43] Jha R., Chablat D., Rouillier F., Moroz G., An Algebraic Method to Check the Singularity-Free Paths for Parallel Robots, In: *Proceedings of ASME 2015 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Boston, Massachusetts, USA, Vol. 5C, pp. 1–10, (2015). 6, 98
- [44] Caro S., Chablat D., Lemoine P., Wenger P., Kinematic Analysis and Trajectory Planning of the Orthoglide 5-Axis, *Proceedings of the ASME 2015 , International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Aug 2015, Boston, United States, (2015). 98, 102
- [45] Symetrie, SYMETRIE - Hexapode, Metrologie, Positionnement, Mesures, [http://http://www.symetrie.fr/fr/](http://www.symetrie.fr/fr/), Retrieved February 16, (2016). 98
- [46] Dspace, dSPACE - DS1103 PPC Controller Board, from www.dspace.com/en/pub/home/products/hw/singbord/ppconbo.cfm, Retrieved February 16, (2016). 88, 98
- [47] Chablat D., Wenger Ph., Working Modes and Aspects in Fully-Parallel Manipulator, *Proceeding IEEE International Conference on Robotics and Automation*, pp. 1964–1969, (1998). 11, 19, 20, 21, 51, 73, 100
- [48] Manubens M., Moroz G., Chablat D., Wenger P., Rouillier F., Cusp Points in the Parameter Space of Degenerate 3-RPR Planar Parallel Manipulators, *ASME Journal of Mechanisms and Robotics*, Vol. 10(1), (2012). 48, 101

-
- [49] Jha R., Chablat D., Rouillier F., Moroz G., Workspace and Singularity analysis of a Delta like family robot, In *Robotics and Mechatronics*, pringer International Publishing, pp. 121-130, (2006). 101
- [50] Maple 2015, Maple User Manual, Toronto: Maplesoft, a division of Waterloo Maple Inc., (2005-2015) 109
- [51] Zhao J-S., Zhou K., Feng Z.J., A theory of degrees of freedom for mechanisms, *Mechanism and Machine Theory*, Volume 39, Issue 6, pp. 621-643, (2004).
- [52] Lazard, D., and Rouillier, F. Solving parametric polynomial systems. *Journal of Symbolic Computation*, Vol. 42 No. 6 : 636 - 667, (2007). 15, 26, 39, 74
- [53] Liang, S., Gerhard, J., Jeffrey, D. J., and Moroz G., "A Package for Solving Parametric Polynomial Systems." *ACM Communications in Computer Algebra*, Vol. 43 No. 3 : 61 - 72 (2009). 26
- [54] Cylindrical Algebraic Decomposition, Maple, www.maplesoft.com/support/help/maple/view.aspx?path=RegularChains%2FSemiAlgebraicSetTools%2FCylindricalAlgebraicDecompose. 28
- [55] Cell Decomposition, <http://www.maplesoft.com/support/help/Maple/view.aspx?path=RootFinding/Parametric/CellDecomposition> 28
- [56] Moroz, G. Sur la decomposition réelle et algébrique des systèmes dépendant de paramètres, Ph.D. thesis, l'Université de Pierre et Marie Curie, Paris, France. (2008). 15, 39, 74
- [57] Sefrioui J., Gosselin C., Singularity analysis and representation of planar parallel manipulators, *Robots and autonomous Systems* 10, pp. 209-224, (1992). 16
- [58] Zlatanov D., Bonev I., and Gosselin C., A Singularities as C-Space Singularities, *Advances in Robot Kinematics: Theory and Applications*, pp. 183-192, Kluwer Academic Publishers, (2002).
- [59] Borrel P., A study of manipulator inverse kinematic solutions with application to trajectory planning and workspace determination, *Proc. IEEE Int. Conf on Rob. And Aut.*, pp. 1180-1185, (1986). 20, 42
- [60] Wenger Ph., Chablat D., Uniqueness Domains in the Workspace of Parallel Manipulators, *IFAC-SYROCO*, Vol. 2, pp 431-436, (1997). 21, 41, 51
- [61] Innocenti C., Parenti-Castelli V., Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators, *Robotics, Spatial Mechanisms and Mechanical Systems*, ASME (1992). 21, 22, 24
- [62] Hunt K.H., Primrose E.J.F., Assembly configurations of some In-parallel-actuated manipulators, *Mechanism and Machine Theory*, 28(1), pp.31-42, (1993). 25, 52
- [63] McAree P.R., Daniel R.W., An explanation of never-special assembly changing motions for 3-3 parallel manipulators, *International Journal of Robotic Research*, 18 (6), pp. 556-574., (1999). 25

- [64] Zein, M., Wenger, P., Chablat, D.: Singular curves in the joint space and cusp points of 3-RPR parallel manipulators. *Robotica* 25(6), 717-724 (2007). 25, 44
25
- [65] Husty, M., Schadlbauer, J., Caro, S., Wenger, P.: The 3-RPS manipulator can have nonsingular assembly-mode changes. In: *Computational Kinematics, Mechanisms and Machine Science* Vol. 15, pp. 339-348. Springer (2014). 25
- [66] Gogu, G., 2008. *Structural synthesis of parallel robots*. Dordrecht: Springer. 12, 16
- [67] Bamberg, H., Wolf, A., Shoham, M.: Assembly mode changing in parallel mechanisms. *IEEE Transactions on Robotics* 24(4), 765-772 (2008). 25
- [68] Macho, E., Altuzarra, O., Pinto, C., Hernandez, A.: Transitions between multiple solutions of the direct kinematic problem. In: *Advances in Robot Kinematics: Analysis and Design*, pp. 301-310. Springer (2008). 25, 44, 52
- [69] Merlet, J.P. and Gosselin, C., 2008. Parallel mechanisms and robots. In *Springer Handbook of Robotics* (pp. 269-285). Springer Berlin Heidelberg. Vancouver 12
- [70] Zein M., Wenger P. Chablat D., Non-Singular Assembly-mode Changing Motions for 3-RPR Parallel Manipulators, *Mechanism and Machine Theory*, Vol 43/4, pp. 480-490, (2008). 25, 44, 68
- [71] Coste M., Chablat D., Wenger Ph., Nonsingular Change of Assembly Mode Without any Cusp, *Advances in Robot Kinematics*, Springer, pp. 105-112, (2014). 25
- [72] Merlet J-P., *Parallel robots*, Springer, Vol. 74, (2001). 12, 26
- [73] Collins GE., Quantifier elimination for real closed fields by cylindrical algebraic decomposition, In *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern*, Springer Berlin Heidelberg, pp. 134-183, (1975). 44, 61
- [74] Bohigas, O. and Manubens, M. and Ros, L.: A complete method for workspace boundary determination on general structure manipulators. *IEEE Transactions on Robotics*, 2Vol. 8/5, pp. 993-1006, (2012). 32
- [75] Moroz G., Rouiller F., Chablat D., Wenger P., On the determination of cusp points of 3-RPR parallel manipulators, *Mechanism and Machine Theory*, Vol. 45(11), pp. 1555-1567, (2010). 74
- [76] Pashkevich, A. and Chablat, D. and Wenger, P.: Kinematics and workspace analysis of a three-axis parallel manipulator: the Orthoglide. *Robotica*, Vol. 24/1, pp. 39-49, (2006). 32, 77
- [77] Chablat D., Wenger P., Séparation des solutions aux modèles géométriques direct et inverse pour les manipulateurs pleinement parallèles, *Mechanism and Machine Theory*, Vol 36/6, pp. 763-783, (2001). 21, 52, 62
- S. Briot and V. Arakelian, Optimal force generation in parallel manipulators for passing through the singular position, *The International Journal of Robotics Research* 27 (8), pp. 967-983, (2008).

-
- [78] Hebsacker, M. and Treib, T. and Zirn, O. and Honegger, M.: Hexaglide 6 DOF and Triaglide 3 DOF Parallel Manipulators. In: *Parallel kinematic machines: theoretical aspects and industrial requirements*, pp. 345-355, (1999). 32
- [79] Husty M., Schadlbauer J., Caro S., Wenger P., Non-Singular Assembly Mode Change of 3-RPS Manipulators, *Proceedings of CK2013, International Workshop on Computational Kinematics, Barcelone*, (2013). 35, 41, 52, 68
- [80] Babu S. R., Raju V. R., Ramji K., Design for optimal performance of 3-RPS parallel manipulator using evolutionary algorithms, *Transactions of the Canadian Society for Mechanical Engineering*, Vol. 37, No. 2, (2013). 35
- [81] Gallardo J., Orozco H., Rico J., Aguilar C., and Perez L., Acceleration analysis of 3-RPS parallel manipulators by means of screw theory, in *Parallel Manipulators, New Developments* (J.-H. Ryu, ed.), I-Tech Education and Publishing, (2008). 35
- [82] Zhou W.-Y., Chen W.-Y., Han X.-G. and Zou F., The Kinematics Algorithm of 3RPS Mechanism with Compound Sphere Joint, *Journal of Harbin Institute Of Technology (New Series)*, 20(6):91-96, (2013). 35
- [83] Cox, D.A. and Little, J. and O'Shea, D.: *Using Algebraic Geometry*, 2nd edition, Graduate Texts in Mathematics. Springer, (2004). 44, 53, 73
- [84] Wenger P., A new general formalism for the kinematic analysis of all nonredundant manipulators, *IEEE Robotics and Automation*, pp. 442-447, (1992). 21, 62
- [85] Moore R. E., Kearfott R. B., Cloud, M. J., *Introduction to interval analysis*, Siam, (2009). 81
- [86] N. Rakotomanga, D. Chablat and S. Caro, Kinetostatic performance of a planar parallel mechanism with variable actuation, In *Advances in robot kinematics: Analysis and design*, pp. 311-320. Springer Netherlands, (2008). 38, 84, 86
- [87] O. Alba-Gomez, P. Wenger and A. Pamanes, Consistent kinetostatic indices for planar 3-DOF parallel manipulators, application to the optimal kinematic inversion, In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 765-774. American Society of Mechanical Engineers, (2005). 84
- [88] S. Caro, D. Chablat, P. Wenger and X. Kong, Kinematic and dynamic modeling of a parallel manipulator with eight actuation modes, In *New Trends in Medical and Service Robots*, pp. 315-329. Springer International Publishing, (2014). 84, 94
- [89] K. Belda and P. Pisa, Homing, calibration and model-based predictive control for planar parallel robots, In *Proceedings of the UKACC International Conference on Control*, (2008). 84
- [90] M. Palpacelli, M. Callegari, L. Carbonari and G. Palmieri, Multibody Analysis and Design of a Reconfigurable Parallel Kinematics Manipulator, In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V009T07A076-V009T07A076. American Society of Mechanical Engineers, (2015). 86

- [91] X. Kong, J. Yu and D. Li, Reconfiguration Analysis of a 2-DOF 3-4R Parallel Manipulator With Planar Base and Platform, In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. V05CT08A034-V05CT08A034. American Society of Mechanical Engineers, (2015). 86
- [92] W. Ye, Y. Fang and S. Guo, Reconfigurable parallel mechanisms with planar five-bar metamorphic linkages, *Science China Technological Sciences* 57, no. 1 : pp. 210-218, 2014. 86
- [93] G. Yang, I-M. Chen, W. K. Lee and S. H. Yeo, Self-calibration of three-legged modular reconfigurable parallel robots based on leg-end distance errors, *Robotica* 19, no. 02, pp. 187-198, (2001). 86
- [94] Neagoe, M., Gogu, G. and Diaconescu, D., 2004. High degree accuracy modelling and calibration of serial robots with large errors. In *Product Engineering* (pp. 397-408). Springer Netherlands. 98
- [95] I-M. Chen, , G. Yang, W. K. Lim and S. H. Yeo, Self-Calibration of Three-Legged Modular Reconfigurable Parallel Robots Based on Measurement Residues, In *Proc. 2nd Intern. Workshop on Computational Kinematics*, pp. 117-132, (2002). 86
- [96] D. Gan, J. S. Dai, J. Dias and L. D. Seneviratne, Variable Motion/Force Transmissibility of a Metamorphic Parallel Mechanism With Reconfigurable 3T and 3R Motion, In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. V05CT08A037-V05CT08A037. American Society of Mechanical Engineers, (2015). 86
- [97] S. Briot and V. Arakelian, Optimal force generation in parallel manipulators for passing through the singular position, *The International Journal of Robotics Research* 27 (8), pp. 967-983, (2008). 95
- [98] S. Caro, D. Chablat, and Y. Hu, Algorithm for the actuation mode selection of the parallel manipulator NAVARO, In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. V05BT08A063-V05BT08A063. American Society of Mechanical Engineers, (2014). 87
- [99] C. Gosselin and J-P. Merlet, On the direct kinematics of planar parallel manipulators: special architectures and number of solutions, *Mechanism and Machine Theory*, Vol. 29, No. 8, pp. 1083-1097, (1994). 89, 90
- [100] NaVARo, Singular assembly mode changing trajectory, <https://www.youtube.com/watch?v=e3uMXA2tapM>, Retrieved February 16, (2016). 95