TECHNISCHE UNIVERSITÄT WIEN

DISSERTATION

Design and Path Optimization of Linear Pentapods Based on the Geometry of their Singularity Varieties

Verfasser:
Arvin RASOULZADEH

Leitender Supervisor:
Dr. Georg NAWRATIL
2te Supervisor:
Dr. Hannes KAUFMANN
3te Supervisor:
Dr. Florian RIST

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften

im

Center for Geometry and Computational Design
Institut für Diskrete Mathematik und Geometrie

Wien, am July 21, 2020

Eigenhändige Unterschrift

Arvin Rasoulzadeh

Declaration of Authorship

I, Arvin RASOULZADEH, declare that this thesis titled, “Design and Path Optimization of Linear Pentapods Based on the Geometry of their Singularity Varieties” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Arvin Rasoulzadeh

Date: 21.7.2020

"If you can't explain something to a six-year-old, you really don't understand it yourself."

Richard Feynman

TECHNISCHE UNIVERSITÄT WIEN

Abstract

Center for Geometry and Computational Design
Institut für Diskrete Mathematik und Geometrie

technischen Wissenschaften

Design and Path Optimization of Linear Pentapods Based on the Geometry of their Singularity Varieties

by Arvin RASOULZADEH

The main goal of this work is to optimize the design and path of linear pentapods via studying the geometrical properties of their related singularity varieties. Several mathematical tools are borrowed to perform the aforementioned tasks. In the first chapter, these tools are briefly reviewed in two groups of *differential geometry* and *algebraic geometry*. The chapter is peppered with several purposeful examples which are given in such a way that they would give an intuition to the reader for the coming higher dimensional counterparts. The second chapter is mainly centred around the algebraic concepts of this study, namely, the *rational parametrization* of the general linear pentapods' singularity variety, *topological and geometrical properties of the singularity variety* and most importantly introducing a new class of linear pentapods under the name of *simple pentapods* which possess a simple-structured singularity variety. The third chapter views the singularity variety from a rather mathematically different discipline, namely the metric spaces. In this chapter, ideas such as *pedal points* on the singularity variety and *object-oriented metric* are investigated. The chapter ends with the main result on the generic number of pedal points on the simple pentapods' singularity variety. The fourth chapter focuses on the concept of variational path optimization by assuming that an already singularity-free path between two non-singular poses of the simple pentapod is given. Combining the tools from chapter one with the results of chapter three paves the way for obtaining a *real time path optimization*. Additionally, the optimized path takes the *base spherical joint* and *prismatic extension* limits into account in such a way that during the related optimized motion the manipulator does not exceed these limits. The chapter comes to an end with a brief introduction of a graphical user interface which provides a user-friendly access to the aforementioned variational path optimization algorithm. Finally, the last chapter names and explains the related unfinished projects in the field. These include *path planning*, *workspace analysis* and the extension of the variational path optimization algorithm to the *platform spherical joints limit* and *leg collision*.

This PhD thesis is based on the following scientific works:

- Rational Parametrization of Linear Pentapod's Singularity Variety and the Distance to It, *Computational Kinematics* [Rasoulzadeh and Nawratil, 2018],
- Linear Pentapods with a Simple Singularity Variety Part I: Determination and Redundant Designs, *IFToMM World Congress on Mechanism and Machine Science* [Rasoulzadeh and Nawratil, 2019a],

- Linear Pentapods with a Simple Singularity Variety Part II: Computation of Singularity-free Balls, *IFToMM World Congress on Mechanism and Machine Science* [Rasoulzadeh and Nawratil, 2019b],

- Variational Path Optimization of Linear Pentapods with a Simple Singularity Variety, *Journal of Mechanism and Machine Theory* [Rasoulzadeh and Nawratil, 2020],

which are all peer reviewed publications.

Examiners: Prof. Gernot Tragler, Dr. Georg Nawratil, Prof. Philippe Wenger.

Acknowledgements

Hereby, I would like to thank my parents for their emotional support through all these years in addition to my brother whom, from time to time, helped me in the areas related to computer science. Finally and above all, my supervisor *Dr. Georg Nawratil*, who deeply promoted my geometric intuition through hours and hours of scientific conversations.

Contents

Declaration of Authorship **iii**

Abstract **vii**

Acknowledgements **ix**

1 Mathematical Frameworks **1**
1.1 Differential Geometry 1
1.1.1 Metric Spaces 1
1.1.2 Analysis on Manifolds 2
1.1.3 Metric Tensor on Smooth Manifolds 3
1.1.4 Geodesics 4
1.1.5 Riemannian Submanifold 5
1.1.6 Orthogonal Projection 5
Cut Locus 5
1.1.7 Metric-dependent Gradients 6
1.1.8 Optimization Techniques and Gradient Descent 7
Quadratic Programming Problem Formulation 7
Gradient Descent Method 8
Computation of Pedal Points 11
1.1.9 Curvature of the Curves 12
1.1.10 Energy of Curves 12
Discretization of Curve Energies 13
1.2 Algebraic Geometry 14
1.2.1 Algebraic Varieties 14
1.2.2 Projective Varieties 16
1.2.3 Critical Values Vs. Singular Points 17
1.2.4 Distance to a Variety via Lagrange Multipliers 18
1.2.5 Rational Varieties 19
1.2.6 Real Algebraic Geometry 22
1.2.7 Singular Value Decomposition 22
1.2.8 Resultants method 23
1.2.9 Gröbner basis method 24
1.2.10 Bézout's Theorem and Extension of Example 18 25

2 Linear Pentapods **27**
2.1 Introduction 27
2.1.1 Review 28
Forward and Inverse Kinematics of the Linear Pentapods 28
Redundancy 28
Singularity Analysis of Linear Pentapods 28
2.1.2 Notation and preparatory work 30
2.2 Singularity Variety of Linear Pentapods 32

2.2.1 Rational Parametrization of Singularity Locus . . . 33
2.3 Designs of the Linear Pentapods Implying a Simple Singularity Variety . . . 34
2.3.1 Linear in Position Variables . . . 34
Planar case . . . 34
Non-planar case . . . 36
2.3.2 Linear in Orientation Variables . . . 37
Planar case . . . 37
Non-planar case . . . 39
2.3.3 Quadratic . . . 40
2.4 Simple Pentapods . . . 41
2.4.1 Σ - Variety . . . 41
2.5 Kinematic Redundant Designs . . . 44
2.5.1 Design 1 . . . 44
2.5.2 Design 2 . . . 45
2.5.3 Design 3 . . . 46

3 Distance to the Singularity Variety **49**
3.1 Object-oriented Metric . . . 49
3.2 General Linear Pentapods . . . 51
3.2.1 Positions (Fixed Orientation Case) . . . 51
3.2.2 Orientations (Fixed Position Case) . . . 51
3.2.3 Euclidean Motion . . . 53
3.2.4 Equiform Motions . . . 55
3.3 Simple Pentapods . . . 55
3.3.1 Linear in Position Variables (LP) . . . 56
Positions (Fixed Orientation Case) . . . 56
Orientations (Fixed Position Case) . . . 56
Euclidean Motion . . . 57
Equiform Motion . . . 58
3.3.2 Linear in Orientation Variables (LO) . . . 58
Positions (Fixed Orientation Case) . . . 59
Orientations (Fixed Position Case) . . . 59
Euclidean Motions . . . 60
Equiform Motions . . . 60
3.3.3 Summary of Results . . . 61

4 Variational Path Optimization of Simple Pentapods **63**
4.1 Preparatory Work . . . 63
4.1.1 Distance to Σ-variety & Modified Orthogonal Projection . . . 63
Pedal points on Σ_3 . . . 63
Pose points on cut locus . . . 64
Modified orthogonal projection . . . 65
4.1.2 Cost Function . . . 65
4.1.3 Step Size . . . 67
4.1.4 Objective Function . . . 67
4.1.5 Remarks on the Curve's Energies . . . 68
4.2 Orthogonal Projection into the Configuration Space . . . 68
4.2.1 Projection on Cylinder's Tangents . . . 68
4.2.2 Projection from Cylinder's Tangents to the Cylinder . . . 70
4.3 Finite Singularity-free Cover . . . 70
4.3.1 Inclusion of Breakpoints . . . 70

4.3.2 Exclusion of Breakpoints 71
4.4 Joint Analysis 71
4.4.1 Prismatic Joint Limits 72
4.4.2 Base Spherical Joint Limits 72
4.5 Variational Path Optimization Algorithm's Manual 73
4.5.1 Input 73
Design parameters 73
Optimization parameters 73
4.5.2 Process I 74
4.5.3 Decision I 74
4.5.4 Process II 74
4.5.5 Decision II 75
4.5.6 Process III 75
4.5.7 Output 75
4.6 Results & Discussions 75
4.6.1 Example I 75
4.6.2 Example II 77
4.6.3 Example III 78
4.6.4 Variational Path Optimization Through Graphical User Interface (GUI) 79
GUI for the Variational Path Optimization of Quadrics 80
GUI for the Variational Path Optimization of Simple Pentapods . . . 82

5 Conclusion and Future Works **85**
5.1 Analytical Path Planning of the Simple Pentapods 85
5.2 Workspace Analysis of Simple Pentapods 87
5.3 Leg Collision Analysis and Platform Spherical Joints Limits 88

A Derivation of Pentapod's Jacobian Matrix **91**

B Variational Path Optimization Flowchart **93**

C Computational Details of the Linear System of Eq. 4.6 **95**
C.1 Matrix of Coefficients **A** 95
C.2 Matrix of Constants **b** 96
C.3 Matrix of Variables **u** 98

Bibliography **99**

List of Figures

1.1 Illustrations of the wrapping *6-patch* parametrization of a sphere where its two defining parameters each sweep the intervals $[-0.45, 0.45]$ (left), $[-0.65, 0.65]$ (middle) and $[-0.85, 0.85]$ (right). 3

1.2 Illustrations of polar parametrization of a sphere. The yellow half-plane depicts $\mathscr{P}$. 4

1.3 Illustrations of the profit per population data for each city (red crosses). The blue line depicts the *linear regression* fit. 8

1.4 Illustrations of the cost function surface (left) and the contour showing the minimum (right). 9

1.5 Illustrations of the convergence of u_c to p^*. Starting with an initial guess u_c, at each iteration u_c is updated to u_+ by moving in the negative direction of the $\nabla f(u_c)$ with a certain positive step size magnitude s on the $T_{u_c}(M)$. Then u_+ is orthogonally project from $T_{u_c}(M)$ into M and the whole process is repeated till convergence is obtained. Informally, one should expect convergence to occur for sufficiently small s. However, in later chapters we approach the step size from a more intelligent point of view. 10

1.6 Gradient flow with respect to the canonical (top-left) and induced (top-right) gradient fields. Below: the corresponding paths on the parameter domain's level sets. The courtesy of the figures goes to Georg Nawratil. 11

1.7 Illustration of a *proper* variation. 12

1.8 Illustrations of affine varieties. Left: a one-sheet hyperboloid $\mathbf{V}(x^2+y^2-z^2-1)$ as a smooth connected surface. Middle: a cone $\mathbf{V}(x^2+y^2-z^2)$ as a non-smooth connected surface. Right: a two-sheet hyperboloid $\mathbf{V}(x^2+y^2-z^2+1)$ as a smooth non-connected surface. 14

1.9 Illustrations of the twisted cubic. Left: The twisted cubic as the intersection $\mathbf{V}(y-x^2) \cap \mathbf{V}(z-x^3)$. Right: A general view. 15

1.10 Illustration of the intersection of lines in the projective plane. The black lines meet on the line at infinity while the red and the blacks meet in the finite plane $\mathbb{R}^2$. 17

1.11 Illustration of distance of a point in space (green) to a curve with cusp singularity. 18

1.12 Illustrations of the comparison of three rational parametrizations of a one-sheet hyperboloid. Left: Rational parametrization using a bundle of line with apex on the plane at infinity. Middle: Rational parametrization using Weierstrass substitution. Right: Rational parametrization of the one-sheet hyperboloid as a ruled surface. The blue silhouette is the hyperboloid that is to be covered by the coloured parametrized surfaces. 20

1.13 Illustration of the intersection of three algebraic varieties (given in blue, green and salmon) in two points (red spheres). The courtesy of the related polynomials goes to Franz Winkler. 23

2.1 Sketch of a linear pentapod. 27

2.2 Illustrations of the counter examples for the necessity of the condition given in Lemma 4. 31

2.3 Red line stands as the linear pentapod's motion platform, while green and black stand for the triangle in the base and the legs respectively. 32

2.4 Geometric interpretation of the conditions yielding a singularity polynomial, which is linear in position variables for a linear pentapod with a planar base: architecturally singular case (left) and the non-architecturally singular case (right). 35

2.5 Geometric interpretation of the conditions yielding a singularity polynomial, which is linear in orientation variables for a linear pentapod with a planar base: architecturally singular case (left) and the non-architecturally singular case (right). 38

2.6 Three possible designs mentioned in Theorem 5. In this figure, a possible design space is illustrated where the point X can be chosen on the depicted bundle of lines. Blue points stand as the possible designs containing three collinear base points and red points as the design with three collapsed spherical joints while the only green point describes the design with four base points on a line. The collinear base points are indicated by red beams. It is noteworthy that illustrated designs are obtained based on the number of co-incidents (which for red, blue and green are 2,1 and 0 respectively). Additionally, it can be shown by a series of Δ-transforms [Borràs, Thomas, and Torras, 2010b], that the singularity loci of all the three cases are identical. . . 39

2.7 One way to have an imaginative illustration of the Σ-variety is to consider a case with a certain similarity in $\mathbb{R}^3$. In order to do so, consider the variety $\mathbf{V}\left(z\left(x^2-y^2-z^2\right)\right)$ which is in fact the union of a plane Σ_1 and a hyperbolic paraboloid Σ_2. In that way, the $\Sigma_1 \cap \Sigma_2$ will be of co-dimension 2 while Σ_1 and Σ_2 are tangent at a 0 dimensional subvariety. Note that due to the fact that in this example we are restricted to $\mathbb{R}^3$ it is impossible to show the 2-dimensional singular set on Σ_2 as it is of co-dimension 4. 42

2.8 Illustration of the kinematic redundant linear pentapod of Section 2.5.1 with a linear singularity variety in position variables. 45

2.9 Illustration of the kinematic redundant linear pentapod of Section 2.5.2 with a linear singularity variety in orientation variables. The suggested design, where the upper part is mounted on the ceiling, can be of interest for e.g. the milling of an object without any need of its repositioning, as the manipulator can go around the object by 360 degrees. 46

2.10 Illustration of the kinematic redundant linear pentapod of Section 2.5.3 with a linear singularity variety in orientation variables. This design also allows a milling by 360° degrees around the object. Moreover, detailed views of the double joint $m_1 = m_2$ at the platform and the circular slider of M_2 are provided. 47

3.1 Illustration of a linear pentapod with a planar base. From a geometrical point of view, the value of Eq. 3.2, for the two poses of the planar pentapod, is equal to $1/5$ of the sum of the squared lengths of the dashed lines between two platforms configurations. 50

3.2 Illustrations are done for $\mathfrak{G} = \left(\frac{3}{5}, \frac{4}{5}, 0, 2, 3, 4\right)$ of the linear pentapod displayed in Fig. 2.1. Fixed orientation (Left): $\mathscr{O}$ has only four real solutions where the closest one $\mathfrak{O} = \left(\frac{3}{5}, \frac{4}{5}, 2.4774, 2.6978, 0.9626\right)$ has a distance of 3.9444 units. Fixed position (right): $\mathscr{P}$ has only two real solutions where the closest one $\mathfrak{P} = (0.3701, 0.5523, 0.7468, 2, 3, 4)$ has a remote distance of 48.4178°. . . . 52

3.3 Given is the pose $\mathfrak{G}$ (black) of the linear pentapod. Left: The closest singular configurations in the position/orientation workspace are given by the pose $\mathfrak{P}$ (yellow) and $\mathfrak{O}$ (blue), respectively. Right: $\mathfrak{M}$ (red) is the closest singular pose under Euclidean motions of ℓ and $\mathfrak{N}$ (yellow) is the closest singularity under equiform motions of ℓ. 53

3.4 Left: For fixed orientation the unique pedal point has coordinates $(\frac{61}{33}, \frac{38}{33}, \frac{92}{33}) \in \mathbb{R}^3$ which has a distance of 1.21854359 units from the given position. Middle-left: For fixed position the four pedal points are illustrated, where the one with coordinates $(0.12661404, 0.81506780, 0.56536126) \in \mathbb{R}^3$ is closest to the given orientation. The corresponding spherical distance equals 15.75049156°. The second pedal point is antipodal to the first one and the spherical distance is the supplementary angle. Middle-right: Illustration of the pose $\mathfrak{G}$ (green) of the linear pentapod studied in Section 3.3.1. The closest singular configurations in the position/orientation workspace are given by the pose $\mathfrak{P}$ (yellow) and $\mathfrak{O}$ (blue), respectively. Right: $\mathfrak{M}$ (red) is the closest singular pose under Euclidean motions of ℓ and $\mathfrak{N}$ (yellow) is the closest singularity under equiform motions of ℓ. 56

3.5 Left: For fixed orientation there exits for each of the two planes a unique pedal point. One pedal point has coordinates $(-8/17, 9/17, 12/17)$ and a distance of 4.80196038 units to the given position and the other pedal point has coordinates $(2, 3, 0)$ and a distance of 4 units. Middle-left: For fixed position the two pedal points are illustrated, where the one with coordinate $(0.11346545, 0.47007115, 0.87530491)$ is closest to the given orientation. The corresponding spherical distance equals 20.82450533°. Middle-right: Illustration of the pose $\mathfrak{G}$ (green) of the linear pentapod studied in Section 3.3.2. The closest singular configurations in the position/orientation workspace are given by the pose $\mathfrak{P}$ (yellow) and $\mathfrak{O}$ (blue), respectively. Right: $\mathfrak{M}$ (red) is the closest singular pose under Euclidean motions of ℓ and $\mathfrak{N}$ (yellow) is the closest singularity under equiform motions of ℓ. 59

4.1 Illustrations of overshooting over the cut locus. On the left, if the points of an initial curve (blue) tend to increase their distance to the ellipse by travelling along the *normal lines* it is expected to witness the breakpoints to first land on the medial axis and then all pile up at the center of the ellipse. But in practice, due to the *jump* of the corresponding pedal point, an overshooting takes place. On the right, it is expected to see the breakpoints to land on the medial axis and slowly go to infinity. However, in practice, the break points oscillate horizontally and slowly move vertically. 64

4.2 Left: The green vector shows the optimized direction for the vector $u - p$. The projection on the 1st, 2nd and 3rd gradient lines are indicated by red, pink and cyan respectively. Note that, one can check the correctness of the optimized move as the green direction refers to moving closer to the medial axis and increasing distance to the parabola eventually. Right: Depending on the geometrical structure of the obstacle, geodesic and bending weight factors, the algorithm may not have a final result. The initial path (blue) variates (green) along the parabola's medial axis till infinity under zero geodesic and bending weight inputs. 66

4.3 Left: Illustration of the pull effect of geodesic energy on the breakpoints. The pattern of the breakpoints preserves a regular shape after variation. Right: A geometrical visualization of the errors. The blue boxes describe the calculated value for the geodesic energy before variation, while the green ones depict this energy after the variation. The red arrows connect a breakpoint to its corresponding update. One way to realize the error's occurrence would be through assigning physical units to the calculations, namely *cm*. In that way the the value computed for the geodesic energy will be of physical dimension cm^2 while the distance is *cm*. Both figures are created for the case of the variation of an initial curve $t \mapsto \left(t+1.25, t^3+2\right)$ where $t \in [-1,1]$ with respect to a parabola and with $\lambda = \eta = 10$ as weight factors. 68

4.4 Imaginative illustration of the “generalized cylinder" $\Gamma \subset \mathbb{R}^6$. Left: The blue curve indicates the initial path while the green one stands as the updated curve (not necessarily belonging to Γ). The red dots are in fact the possible imaginative pedal points on the singularity surface Σ. Middle: The projection of the updated curve into the the tangent space. Right: A “realistic" interpretation of such projections in the orientation space. The blue vector shows the original orientation of the pose p^j and the green vector depicts the orientation after update. The cross, x, stands as the back-projection into the “tangent to sphere at p^j", while the red vector describes the final accepted motion after back-projection from the tangent to sphere. 69

4.5 Imaginative illustration of minimal singularity-free cover of the curve. Left: If a segment is not fully covered by singularity-free balls then a breakpoint will be added in the mid of the uncovered part. Right: If two adjacent singularity-free balls are covering a breakpoint that breakpoint is removable. Note that such a removal should be in such a way that it does not alter the cover for adjacent to the neighboring breakpoints. 70

4.6 Left: Illustration of a simple pentapod with the LP-property in addition to a magnified view of the base spherical joints. Base joint limit is modelled by a cone with an axis orthogonal to the base plane. Right: Imaginative illustration of the projection on a tangent of a *Q*-variety. 71

4.7 Illustration of the paths in $\mathbb{R}^6$ and their corresponding pedals as vector fields along paths in $\mathbb{R}^3$. Top-left: Initial path (blue) and its corresponding pedals (purple), Top-right: Final path (green) and its corresponding pedals (red) (blue objective curve), Middle-left: Modification of the final path under 1st prismatic leg's safe-zone breach (golden objective curve), two golden sphere's portions depict the minimum contraction and maximum extension of the first leg, Middle-right: Modification of final path under 2nd base cone's safe-zone breach (magenta objective curve). Bottom-right: Modification of final path under 2nd base cone's safe-zone breach and 1st prismatic leg's maximum extent safe-zone breach at the same time (cyan objective curve). Bottom-left: Objective function (vertical axis) per number of iterations (horizontal axis) descent. Note that this descent fully makes sense with respect to the geometry of the optimized motions. The animated versions of these figures are downloadable via `http://www.geometrie.tuwien.ac.at/rasoulzadeh/`. 76

4.8 Illustration of changes in number of breakpoints per iteration during the exclusion/inclusion process. The blue curve depicts the case where no joint limit is applied while the golden and purple ones stand as the changes under 1st prismatic joint limit and 2nd base spherical joint limits. The cyan curve (not visible) describing the consideration of 1st prismatic and 2nd cone limit simultaneously is fully covered by the golden curve due to the similarity of change per iteration. 78

4.9 Illustration of the same example as in Fig. 4.7 but allowing the inclusion and exclusion of breakpoints to obtain a minimal singularity-free cover. Bottom-left: The jumps in the objective functions result from the inclusion of breakpoints. Bottom-right: The green breakpoints are not “exactly" on the intersection curve (cyan) due to the fact that a certain small gap for computational insurance has been taken into account 79

4.10 Illustration of the *sawtooth* motion and its smoothed optimized version. Top: Initial path (blue) and its corresponding pedals (purple), Bottom: Final path (green) and its corresponding pedals (red). 80

4.11 The graphical user interface of the variational path optimization of the quadrics. 81

4.12 The graphical user interface of the variational path optimization of the simple pentapods. 82

5.1 Illustration of the workspace in position space under constant orientation $(u_1,u_2,u_3)^T = (1,0,0)^T$. 87

List of Abbreviations

lhs	**l**eft **h**and **s**ide
rhs	**r**ight **h**and **s**ide
DoF	**D**egree **o**f **F**reedom
GUI	**G**raphical **U**ser **I**nterface
LO	**L**inear in **O**rientation Variables
LP	**L**inear in **P**osition Variables

List of Symbols

Name	Symbol
Natural numbers	$\mathbb{N}$
Rational numbers	$\mathbb{Q}$
Real numbers	$\mathbb{R}$
Complex numbers	$\mathbb{C}$
Ring	R
Algebraic variety	$\mathbf{V}$
Generalized cylinder	Γ
Pentapod's singularity variety	Σ
Cost function	$\mathscr{C}$
Objective function	$\mathscr{C}'$
Initial guess i-th coordinate	p_i
Pose i-th coordinate	u_i
Pedal points with respect to the breakpoint p^i on Σ-variety	q^i
Pedal points with respect to the breakpoint p^i on Q-variety	$\mathfrak{q}^i$
Metric function	d
Object-oriented metric function	$\mathfrak{d}$
Orthogonal projection with respect to object-oriented metric	π
Orthogonal projection with respect to Euclidean metric	Pr
Number of breakpoints	n
Number of pedal points with respect to the breakpoint p	m_p
Tensor product of matrices	$\otimes$
Hadamard product of matrices	$\circ$
Metric tensor	g
Metric tensor matrix (with respect to the canonical basis)	$\mathbf{g}$
Metric space / Smooth manifold	M
The collection of all tangent smooth vector fields on M	$\mathfrak{X}(M)$
Tangent space to the manifold M at point p	$T_p(M)$
Quadric normal form	Q^N
Disjoint union	$\dot{\cup}$
Step size (learning rate)	γ

Dedicated to my parents, Anahita and Mahmoud

Chapter 1

Mathematical Frameworks

The main goal of this chapter is to briefly review the mathematical tools used throughout the thesis. The chapter is divided into two parts. The first part, under the title of *differential geometry*, mainly reviews the differential geometric concepts related to the study of the mechanical objects coming in the later chapters. However, the section contains topics that are technically not necessarily differential geometric (i.e. *optimization techniques*) but are still included there as they are mainly viewed from a differential geometric point of view. The second section comes with the title of *algebraic geometry* and revisits some general but useful concepts. This section is specifically important in the later chapters as it gives the computational and algebraic tools necessary to study certain classes of surfaces related to the mechanical objects under study.

Finally, it is noteworthy that most of the examples presented in this chapter are designed purposefully for the lower dimensions, in which the reader can visualize the related geometric objects. In this way the reader would have an intuition about the higher dimensional counterparts to come.

1.1 Differential Geometry

Differential geometry is a mathematical discipline that uses techniques of differential calculus, integral calculus, linear and multilinear algebra to study problems in geometry. The theory of plane and space curves and surfaces in the three-dimensional Euclidean space formed the basis for the developments of differential geometry during the 18th and 19th century.

Since the late 19th century, differential geometry has grown into a field concerned more generally with the geometric structures on differentiable manifolds. Differential geometry is closely related to differential topology and the geometric aspects of the theory of differential equations.

Here, the goal is to review those concepts of differential geometry which in later chapters are used to study the geometry of the kinematic structures under investigation.

1.1.1 Metric Spaces

In 1906, Maurice Fréchet introduced the metric spaces in his work *Sur quelques points du calcul fonctionnel*. However the name is chosen by Felix Hausdorff [Fréchet, 1906]. A metric space is a set together with a metric on the set. The metric is a function that defines a concept of *distance* between any two members of the set, which are usually called points. In the arguments to come, metric spaces play an important role, hence in order to have a clear view, a formal definition is of essence.

Definition 1. *A “metric space" is an ordered pair (M,d) where M is a set and d is a metric on M, i.e., a function*

$$d : M \times M \longrightarrow \mathbb{R} \tag{1.1}$$

such that for any $x, y, z \in M$, the following holds:

1. $d(x,y) = 0 \Longleftrightarrow x = y$,
2. $d(x,y) = d(y,x)$,
3. $d(x,z) \leq d(x,y) + d(y,z)$.

The function d is also called "distance function" [Rudin et al., 1964, page 38].

Example 1. *The followings are some well-known examples in the literature:*

- *Any normed vector space is a metric space by defining $d(x,y) = \|x - y\|$,*
- *If (M,d) is a metric space and X is a subset of M, then (X,d) becomes a metric space by restricting the domain of d to $X \times X$,*
- *Given an injective function f from any set A to a metric space (X,d), then $d\left(f(x), f(y)\right)$ induces a metric on A.*
- *For any set there is always a metric associated to it, where $d(x,y) = 0$ if $x = y$ and $d(x,y) = 1$ otherwise. This metric is called "discrete metric".*

1.1.2 Analysis on Manifolds

It was probably the discoveries of Gauss on the curvature of the regular surfaces that founded the subject of differential geometry. After Riemann, the subject was an organized discipline. Many of these discoveries were intrinsic; not depending on how the surface was situated in space. This led to the definition of manifolds [Selig, 2004, page 7-8]. Manifolds are (second countable, Hausdorff) spaces that locally resemble the Euclidean space which allow one to do calculus on them [Boothby, 1986, page 6]. However, globally they are not necessarily isomorphic to $\mathbb{R}^n$. These spaces are covered by local coordinate patches, in such a way that each patch is homeomorphic to an open set in $\mathbb{R}^n$. In the case of *smooth manifolds*, whenever the patches overlap, the two coordinate systems are related to each other by a smooth change of coordinates.
In this context, the maps that we are interested in are the smooth maps between smooth manifolds. A special case of this is a smooth curve. In general, such a curve is seen as a smooth map from an interval of $\mathbb{R}$ into a smooth 1-dimensional manifold M.

Finally, to obtain rigorous mathematical formulation of the above definitions, one could resort to excellent texts such as [Lee, 2003], [O'Neill, 1983] and [Boothby, 1986].

In this part some well-known parametrizations of the sphere are reviewed. These examples will give a visual understanding of the coordinate neighbourhoods and their overlaps. In this thesis, for every manifold M, each coordinate neighbourhood is written as (U, ϕ), where $U \subseteq M$ is an open subset and $\phi : U \longrightarrow \mathbb{R}^n$ is the parametrization.

Example 2. *"6-patch coordinate neighbourhoods" is a routine initial example of the sphere parametrizations. The visualization of these coordinate neighbourhoods gives an intuition of the overlap of the coordinate charts and the diffeomorphic transitions between them. Defining the following notations:*

$$\mathbb{R}_i^+ := \{(x^1, x^2, x^3) \in \mathbb{R}^3 \mid x^i \gneqq 0\}, \tag{1.2}$$

$$\mathbb{R}_i^- := \{(x^1, x^2, x^3) \in \mathbb{R}^3 \mid x^i \lneqq 0\}, \tag{1.3}$$

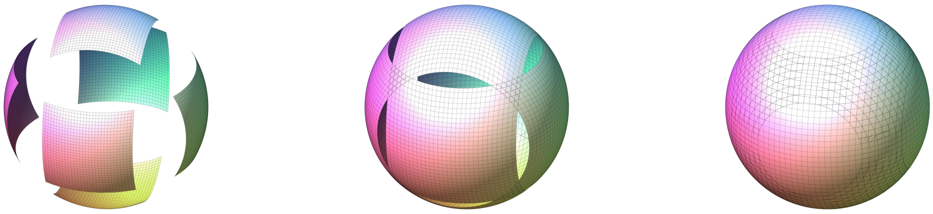


FIGURE 1.1: Illustrations of the wrapping *6-patch* parametrization of a sphere where its two defining parameters each sweep the intervals $[-0.45, 0.45]$ (left), $[-0.65, 0.65]$ (middle) and $[-0.85, 0.85]$ (right).

the 6-patch coordinate neighbourhoods are the half-spheres and are of the following types:

$$\left(\pi_1(x^1,x^2,x^3),\, S^2 \cap \mathbb{R}_1^+\right),\ \left(\pi_2(x^1,x^2,x^3),\, S^2 \cap \mathbb{R}_2^+\right) \tag{1.4}$$
$$\left(\pi_3(x^1,x^2,x^3),\, S^2 \cap \mathbb{R}_3^+\right),\ \left(\pi_1(x^1,x^2,x^3),\, S^2 \cap \mathbb{R}_1^-\right) \tag{1.5}$$
$$\left(\pi_2(x^1,x^2,x^3),\, S^2 \cap \mathbb{R}_2^-\right),\ \left(\pi_3(x^1,x^2,x^3),\, S^2 \cap \mathbb{R}_3^-\right) \tag{1.6}$$

where the map π_i maps the corresponding half sphere to the plane $x^i = 0$ (cf. Fig. 1.1).

Example 3. *The parametrization based on "polar and azimuthal angles", also known as* spherical coordinates, *is another common way of parametrizing the sphere. One of the theoretical main reasons of its use is the fact that it would be helpful in the sense of understanding the concept of manifolds and their related geometric features.*
Fig. 1.2 depicts one of the aforementioned coordinate neighbourhoods, namely

$$\left(S^2 \setminus \mathscr{P},\, \psi\right) \tag{1.7}$$

where

$$\mathscr{P} = \{(x,0,z) \mid x \geq 0\} \tag{1.8}$$

and

$$\psi^{-1}(\theta,\phi) = (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta)). \tag{1.9}$$

1.1.3 Metric Tensor on Smooth Manifolds

Evidently, to do calculation on smooth manifolds an *inner product* is required, where this inner product can be derived from a metric tensor that varies smoothly on the manifold.

Definition 2 (O'Neill, 1983, page 54)**.** *A Riemannian metric tensor on a smooth manifold M is a smooth, symmetric and positive definite $(0,2)$-tensor field g.*

In other words, g assigns a symmetric, positive definite bilinear form:

$$g(p) \approxeq g_p : T_pM \times T_pM \longrightarrow \mathbb{R}. \tag{1.10}$$

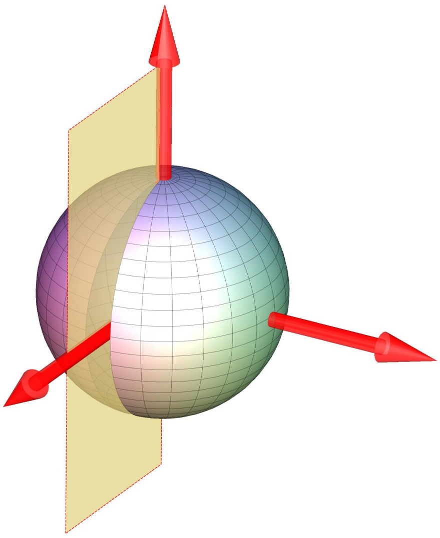


FIGURE 1.2: Illustrations of polar parametrization of a sphere. The yellow half-plane depicts $\mathscr{P}$.

The matrix of g relative to an orthonormal basis $e_1,\ldots,e_m$ of $\mathbb{R}^m$ is denoted by the $m\times m$ matrix $\mathbf{g}$ where its entries are

$$\mathbf{g}_{i,j} = g\left(e_i, e_j\right), \quad 1 \le i, j \le m, \tag{1.11}$$

[O'Neill, 1983, page 50].
The following definition defines the relation between the metric tensor and the metric:

Definition 3. *Consider the metric tensor g. For two points of the Euclidean space $\mathbb{R}^m$ namely, $u=(u_1,\ldots,u_m)$ and $p=(p_1,\ldots,p_m)$ define the metric $d(p,u)$ as:*

$$d(p,u)^2 := C^T.\mathbf{g}.C, \tag{1.12}$$

where $C^T = (u_1-p_1,\ldots,u_m-p_m)^T$. For more details cf. [Boothby, 1986, chapter 2].

Note that the rhs of Eq. 1.12 serves as the desired *inner product*:

$$\langle -,- \rangle : \mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R}, \tag{1.13}$$
$$(X,Y) \longmapsto X^T.\mathbf{g}.Y, \tag{1.14}$$

Finally, $(\mathbb{R}^m, g)$ forms a Riemannian manifold.

1.1.4 Geodesics

Geodesics are one of the two fundamental concepts of Riemannian Geometry. The notion of a geodesic can be thought of as a curve with zero acceleration. From another point of view, one can think of geodesic as a generalization of the Euclidean notion of *straight line* [Do Carmo, 2016, page 60, O'Neill, 1983, page 67].

Remark 1. *The set of all geodesics of a sphere $\mathbf{S}^2$ is the set of all* great circles *(the ones that are obtained by intersecting $\mathbf{S}^2$ and the planes passing through its centre).*

1.1.5 Riemannian Submanifold

A *Riemannian submanifold* N of a Riemannian manifold M is a submanifold of M equipped with the *Riemannian metric* inherited from M. Formally, if (M,g) is a Riemannian manifold and $N \subset M$ is an immersed submanifold, one can define a smooth symmetric 2-tensor $g|_N = \imath^* g$, where $\imath : N \hookrightarrow M$ is the inclusion map. Consequently, for $X,\, Y \in T_p N$ this reads as

$$(g|_N)\,(X,Y) = \imath^* g\,(X,Y) = g\,(\imath_* X, \imath_* Y) = g\,(X,Y)\,. \tag{1.15}$$

This means that $g|_N$ is just the restriction of g to vectors tangent to N [Lee, 2003, page 191] (for the detailed discussion on the immersed submanifolds, *push forwards* $\imath_*$, and *pull backs* $\imath^*$, see [Boothby, 1986, chapter 3]).

Example 4 (Lee, 2003, page 191)**.** *Consider the case of a unit sphere S^2 in 3-dimensional real space with the spherical coordinates. Then the induced metric tensor in matrix format is given by*

$$\mathbf{g} = \begin{pmatrix} 1 & 0 \\ 0 & \sin^2(\theta) \end{pmatrix} \tag{1.16}$$

and is called the "round metric". Hence, $\mathbf{S}^2$ with the round metric becomes a Riemannian submanifold of $\mathbb{R}^3$.

Note that, the round metric can be obtained from the parametrizations described in Example 3.

Lemma 1. *Assume M to be an immersed m-dimensional submanifold of $\mathbb{R}^n$ and p an arbitrary point in $\mathbb{R}^n \setminus M$. Consider $\mathscr{E}$ to be the set of all local extrema of the distance function $d\,(p,M)$ then $q \in \mathscr{E}$ iff $q - p \in T_q\,(M)^{\perp}$.*

Proof. Without loss of generality, assume p is located at the origin. Now, if $q \in \mathscr{E}$ then there exists a parametrization ψ by which $q = \psi(x^1,\ldots,x^m)$ such that $\forall i,\ 1 \le i \le m$ we have

$$0 = \frac{\partial}{\partial x^i} \|\, \psi(x^1,\ldots,x^m)\,\| = 2\,\langle \frac{\partial \psi}{\partial x^i}, \psi \rangle, \quad i = 1\ldots m, \tag{1.17}$$

which gives $\forall\, v \in T_q\,(M)$, $\langle v, \psi \rangle = 0$ and hence $q = \psi(x) \in \mathscr{E}$. The reverse trivially holds. □

1.1.6 Orthogonal Projection

One of the fundamental tools later used for the optimization of the motion paths of the kinematic structures under study is the *orthogonal projection*.

Definition 4 (Dudek and Holly, 1994)**.** *Let M be a non-empty subset of the metric space $\mathbb{R}^n$. The orthogonal projection into M, is the relation $\pi \subset \mathbb{R}^n \times M$ such that for each $p \in \mathbb{R}^n$ we have*

$$\pi\,(p) := \{q \in M \ :\ d\,(p,q) = \rho\,(p,M)\}, \tag{1.18}$$

where d is the metric of $\mathbb{R}^n$ and $\rho\,(p,M) := \inf_{q \in M} d\,(p,q)$.

Remark 2. *π is defined as a relation rather than a map due to the fact that it might not be a function.*

Cut Locus

In Definition 4, we emphasized on π being a relation rather than a function. It turns out such a failure has close ties to the concept of *cut locus* of the metric spaces.

Definition 5 (Petersen, Axler, and Ribet, 2006)**.** *Let M be metric space, equipped with a metric d, and let $p \in M$ be a point. The cut locus of p in M, $CL_p(M)$, is the locus of all points in M for which there exists at least two distinct shortest paths to p. More accurately, $u \in CL_p(M)$ if and only if there exists two paths $c_1, c_2 : I \longrightarrow M$ such that $c_1(0) = p = c_2(0)$, $c_1(1) = u = c_2(1)$, $\|c_1\| = d(u,p) = \|c_2\|$ and the trajectories of the two paths are distinct.*

Example 5. *Let $M = \mathbf{S}^2$ in $\mathbb{R}^3$. Then the cut locus of every point on the sphere consists of exactly another point, namely the antipodal one.*

1.1.7 Metric-dependent Gradients

The usual definition of a gradient of a smooth function has a simplified structure which corresponds to the *ordinary Euclidean metric* and the *canonical basis*, where in the light of Definition 3 is read as

$$\|p-u\|^2 = (p-u)^T . \mathbf{I} . (p-u), \tag{1.19}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix and p and u are two point belonging to $\mathbb{R}^n$. ∇f shows the maximum directional derivative of f and $\|\nabla f\|$ stands as its value. Clearly, the aforementioned concepts rely on the *metric* of the space and hence choosing different metrics gives rise to *different gradients*. Consequently, these facts lead to the definition of a more general gradient operator that have the effect of the metrics encoded in:

Definition 6 (O'Neill, 1983, page 85)**.** *Suppose f is a real-valued function on $\mathbb{R}^n$ and ∇f is the "canonical gradient". Then, the "gradient of f with respect to a positive definite symmetric metric g" (see Section 1.1.3) is defined as follows:*

$$\nabla_g f := \mathbf{g}^{-1} . \nabla f. \tag{1.20}$$

Remark 3. *Assume W to be an inner product space and p and u two points belonging to it. From now on, through the rest of the thesis, we use the following notations:*

$$(p-u)^T . \mathbf{I} . (p-u) = \langle p-u, p-u \rangle = \|p-u\|^2, \tag{1.21}$$

$$(p-u)^T . \mathbf{g} . (p-u) = \langle p-u, p-u \rangle_g = \|p-u\|_g^2, \tag{1.22}$$

where $\mathbf{g}$ is a metric tensor matrix other than the identity matrix ($\mathbf{I}$).

Example 6. *The space $\mathbb{R}^n$ with the ordinary Euclidean metric, gives $\nabla_g f = \mathbf{I}^{-1} . \nabla f = \nabla f = \left(\partial f / \partial x^1, \cdots, \partial f / \partial x^n\right)$ which is the "well known" formula for the gradient.*

Definition 6 describes the concept of the generalized gradient in the Euclidean space $\mathbb{R}^n$. However, one can define the gradient operator on a more general structures, namely, the Riemannian manifolds via operators [O'Neill, 1983, page 85-86]). Here, we resorted to the "less" general definition due to the fact that through out this thesis the geometrical objects that the gradient should operate on are merely the (immersed) submanifolds of $\mathbb{R}^n$. These submanifolds inherit the gradient operator from the ambient Euclidean space. The following Lemma shows that the inherited gradient coincides with the *gradient operator defined on a manifold*.

Lemma 2. *Let f be a smooth function on $\mathbb{R}^n$ and let M be a submanifold of it. Then the gradient of the map f restricted to M at a point $p \in M$, $\left(\nabla f|_M\right)_p$, is the orthogonal projection of $(\nabla f)_p$ onto $T_p(M)$.*

Proof. If g is the *Riemannian metric* on $\mathbb{R}^n$ then $g|_M(\nabla f|_M, X) = X(f|_M)$ for all $X \in \mathfrak{X}(M)$ denoting the set of all smooth tangent vector fields on M (cf. [O'Neill, 1983]). Now, considering the decomposition at the point $p \in M$, $(\nabla f)_p = v_o + v_t$ where $v_t \in T_p(M)$ and $v_o \in T_p(M)^{\perp}$ we have [1]

$$g|_M(\nabla f|_M, X) = X(f|_M) = X(f) = g(\nabla f, X) = g(v_o + v_t, X) = \\ g(v_o, X) + g(v_t, X) = g(v_t, X).$$

which gives

$$g(\nabla f|_M, -) = g(v_t, -). \tag{1.23}$$

□

Remark 4. *Note the in Eq. 1.20, $\mathbf{g}^{-1}$ is an $n \times n$ matrix while ∇f is by default considered to be a $n \times 1$ column matrix. The matrix format is computationally advantageous due to being programming-friendly. In fact in vectorized programmings where tensor product of matrices takes place such presentation is of great help.*

1.1.8 Optimization Techniques and Gradient Descent

Mathematical optimization or mathematical programming is the selection of a best element from a set of available alternatives. Optimization problems arise in many different disciplines such as computer science and engineering to economics. The generalization of optimization theory constitutes a vast area of applied mathematics.

In this thesis the main focus is on a sub-branch of optimization called *quadratic programming*. The quadratic programming is the process of solving the problem of optimization of a quadratic function of several variables subject to linear constraints on these variables which makes this subfield to be a particular type of nonlinear programming [Nocedal and Wright, 2006].

Quadratic Programming Problem Formulation

Suppose that:

- $\mathbf{c}$ is a real-valued n-dimensional vector,
- $\mathbf{b}$ is a real-valued m-dimensional vector,
- $\mathbf{Q}$ is an $n \times n$-dimensional real symmetric matrix,
- $\mathbf{A}$ is an $m \times n$-dimensional real matrix,

then, the quadratic programming problem with n variables and m constraints can be formulated as follows. The objective is to find an n-dimensional vector $\mathbf{u}$, that will

- minimize $\frac{1}{2}\mathbf{u}^T\,\mathbf{Q}\,\mathbf{u} + \mathbf{c}^T\,\mathbf{u}$,
- subject to $\mathbf{A}\,\mathbf{u} \preccurlyeq \mathbf{b}$,

where $\mathbf{u}^T$ denotes the vector transpose of $\mathbf{u}$. However, in the upcoming problems in this thesis the constraints under study are merely the equality and hence we restrict our attention to the case of $\mathbf{A}\,\mathbf{u} = \mathbf{b}$.

[1] In the following equation the subscript p is dropped intentionally.

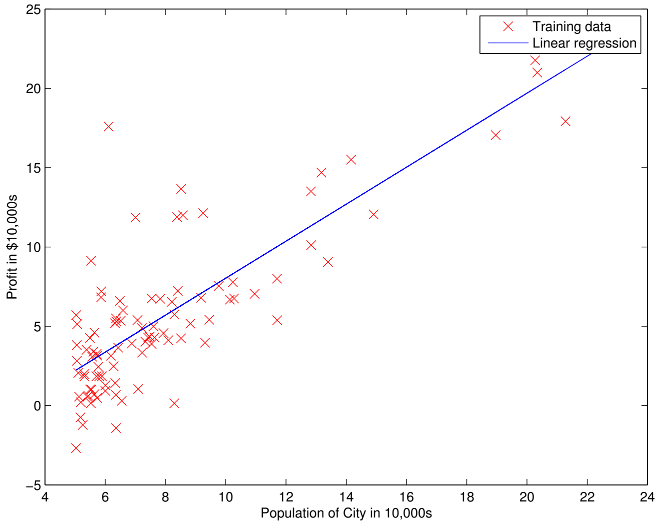


FIGURE 1.3: Illustrations of the profit per population data for each city (red crosses). The blue line depicts the *linear regression* fit.

Gradient Descent Method

Gradient descent is an iterative optimization algorithm for finding a local minimum of a differentiable function [Lemaréchal, 2012].
Gradient descent is based on the observation that if the multivariable function (from now on we call it *cost function*) $F(\mathbf{x})$ is defined and differentiable in a neighbourhood of a point $\mathbf{a}$, then $F(\mathbf{x})$ decreases *fastest* if one goes from $\mathbf{a}$ in the direction of the negative gradient F at $\mathbf{a}$, namely, $-\nabla F(\mathbf{a})$. More accurately, if the *update* is taking place in the following way

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n), \tag{1.24}$$

for $\gamma \in \mathbb{R}^+$ (which is known as *step size* or *learning rate*). If γ is taken *adequate enough* (small enough) then $F(\mathbf{a}_n) \geqq F(\mathbf{a}_{n+1})$. In such a way one can view the decrease of the cost function till convergence which implies the local minimum obtaining.

The following *regression* example gives an intuition regarding the quadratic programming case via gradient descent. In order to focus more on the geometric nature of this optimization case, the numerical details are merely given as points in the Fig. 1.3.

Example 7 (Ng, 2012)**.** *Suppose a CEO of a restaurant franchise considers different cities for opening a new outlet. The chain already has trucks in various cities and the data for profits and population is given for each city via red crosses in Fig. 1.3. We would like to use this data to select which city to expand to. This problem yields a rather regression solution in which the following cost function is proposed:*

$$\mathscr{C}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(u^i\right) - y^i\right)^2, \tag{1.25}$$

where m is the number of data samples and $h_\theta(x)$ is given by the following linear model

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1. \tag{1.26}$$

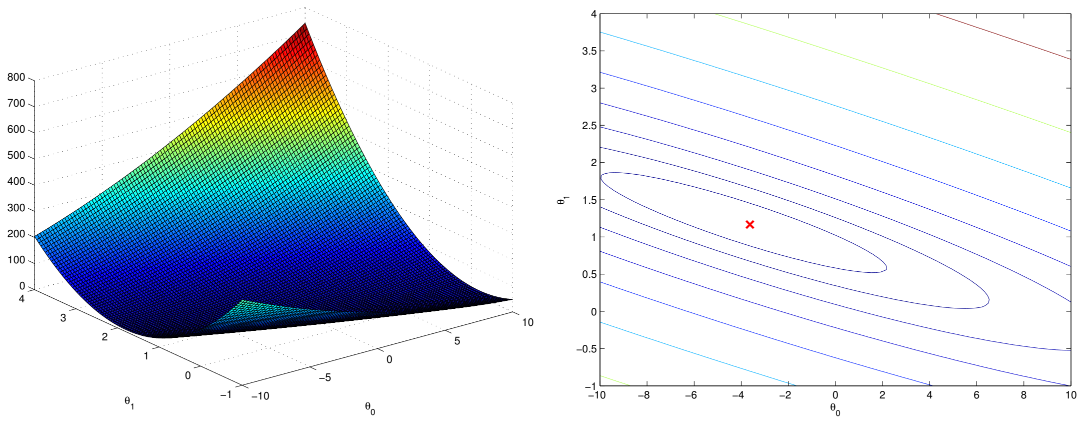


FIGURE 1.4: Illustrations of the cost function surface (left) and the contour showing the minimum (right).

The parameters of our model are θ_j values which we adjust to minimize the cost $\mathscr{C}(\theta)$. Using the gradient descent, each iteration performs the update

$$\theta_j := \theta_j - \gamma \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta \left(x^i \right) - y^i \right) x_j^i, \tag{1.27}$$

by keeping the learning rate $\gamma = 0.01$, with each step of gradient descent, our parameter θ_j come closer to the optimal values that will achieve the lowest cost $\mathscr{C}(\theta)$. Finally, the results are depicted in Fig. 1.4.

Having the following definitions in mind gives a geometrical view towards the problems of a similar nature as Example 7 in higher dimensions.

Definition 7 (O'Neill, 1983, page 20)**.** *A* hypersurface *in a manifold M is a submanifold N whose* codimension, $\dim M - \dim N$, *is 1.*

Definition 8 (Lee, 2003, page 103)**.** *Let $U \subset \mathbb{R}^n$ be an open set, and let $F : U \longrightarrow \mathbb{R}^k$ be a smooth function. The* graph *of F is the subset of $\mathbb{R}^n \times \mathbb{R}^k$ defined by*

$$\Gamma(F) = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^k \ : \ x \in U \text{ and } y = F(x)\}. \tag{1.28}$$

Indeed, in the case of Example 7, the cost function $\mathscr{C}(\theta)$ creates a bowl shaped quadric surface in $\mathbb{R}^3$ which is also a graph surface (see Fig. 1.4). In later chapters, when dealing with a much more complicated cost function in several variables and in higher dimensions, Definitions 7 and 8 along with Example 7 come handy in bearing the geometrical intuition of the aforementioned cost function in mind.
In various geometric problems involving the optimization, one has to minimize a function defined on a Riemannian manifold M. In many cases to be seen in later chapters, M is interpreted as a kinematic space where a path on this manifold describes the poses of a rigid body, orientations or positions. For these applications it is important to introduce a meaningful metric on M with respect to the problem to be solved. Then, one takes gradient with respect to that metric. The following example emphasized on defining such a "meaningful metric" by visualizing the faster convergence on a surface.

Example 8. *Consider a point p in $\mathbb{R}^n$ and an m-dimensional immersed submanifold M. The goal is to compute the closest point $p^* \in M$ to p. By Lemma 1 we know that p^* is the*

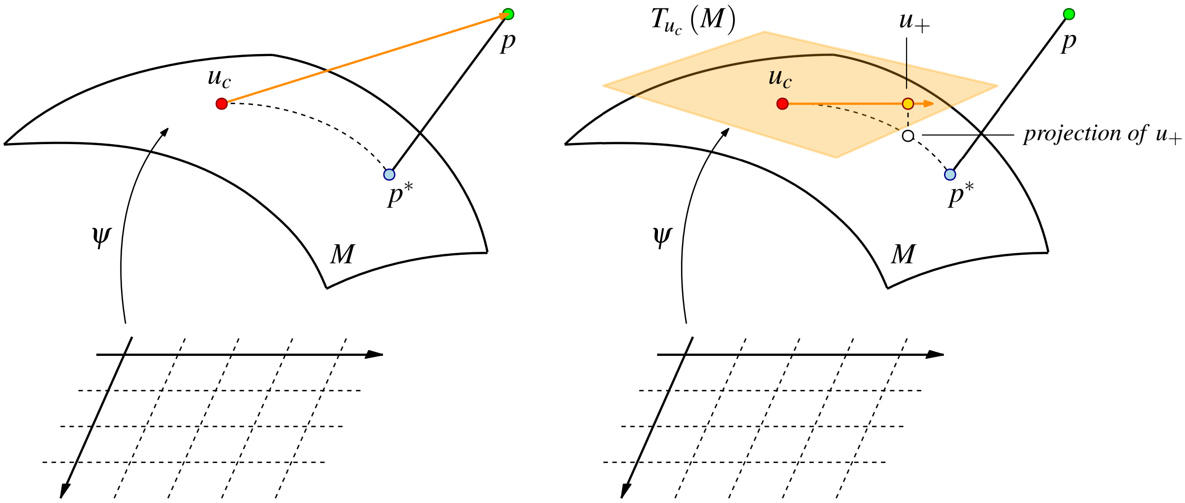


FIGURE 1.5: Illustrations of the convergence of u_c to p^*. Starting with an initial guess u_c, at each iteration u_c is updated to u_+ by moving in the negative direction of the $\nabla f(u_c)$ with a certain positive step size magnitude s on the $T_{u_c}(M)$. Then u_+ is orthogonally project from $T_{u_c}(M)$ into M and the whole process is repeated till convergence is obtained. Informally, one should expect convergence to occur for sufficiently small s. However, in later chapters we approach the step size from a more intelligent point of view.

footpoint of the normal from p to M. p^* can also be seen as the image of an orthogonal projection into M. This problem requires the minimization of the distance function from the point p to M. Since M is a smooth manifold it is covered by the coordinate neighbourhoods. Obtaining the closest point on M requires taking all coordinate neighbourhood into account. Focusing on a coordinate neighbourhood (U,ϕ) of M, one could think of it as a "parametric surface" indicated by $\psi(x^1,x^2,\dots,x^m) := \phi^{-1}(x^1,x^2,\dots,x^m)$. Then the distance function to be minimized is

$$f(x^1,x^2,\dots,x^m) := \frac{1}{2}\|\psi(x^1,x^2,\cdots,x^m)-p\|^2. \tag{1.29}$$

Then the gradient will be

$$\nabla f = \left(\frac{\partial f}{\partial x^1},\dots,\frac{\partial f}{\partial x^m}\right), \tag{1.30}$$

$$\forall 1 \le i \le m, \ \frac{\partial f}{\partial x^i} = \langle \psi(x^1,x^2,\dots,x^m)-p, \frac{\partial \psi}{\partial x^i}\rangle. \tag{1.31}$$

Geometrically, the rhs of Eq. 1.31 implies that the gradient vector ∇f is in fact the projection of $\overrightarrow{qp}$ into the tangent space to M at each point $\psi(x^1,x^2,\dots,x^m) =: q$, namely $T_q(M)$. Consequently, in the light of Lemma 1, $\nabla f = 0$ implies $(p-p^*) \in T_{p^*}(M)^{\perp}$.
Now, let us solve this optimization problem using the gradient descent algorithm. The aforementioned algorithm is formulated in the parameter domain through the following iterative formula (see Fig. 1.5)

$$u_+ = u_c - \gamma\nabla f(u_c). \tag{1.32}$$

where u_c is the current point on the surface, u_+ is the updated point on $T_{u_c}(M)$ and γ is the step size. Eq. 1.32 does not consider the distortion between parameter domain and the actual surface (see Figs. 1.6 - left). In order to take it into account, one has to compute "the gradient with respect to the induced metric on the surface" which is in fact the metric induced on the surface from the ambient space. Such a gradient is computed using Eq. 1.20 and modifies

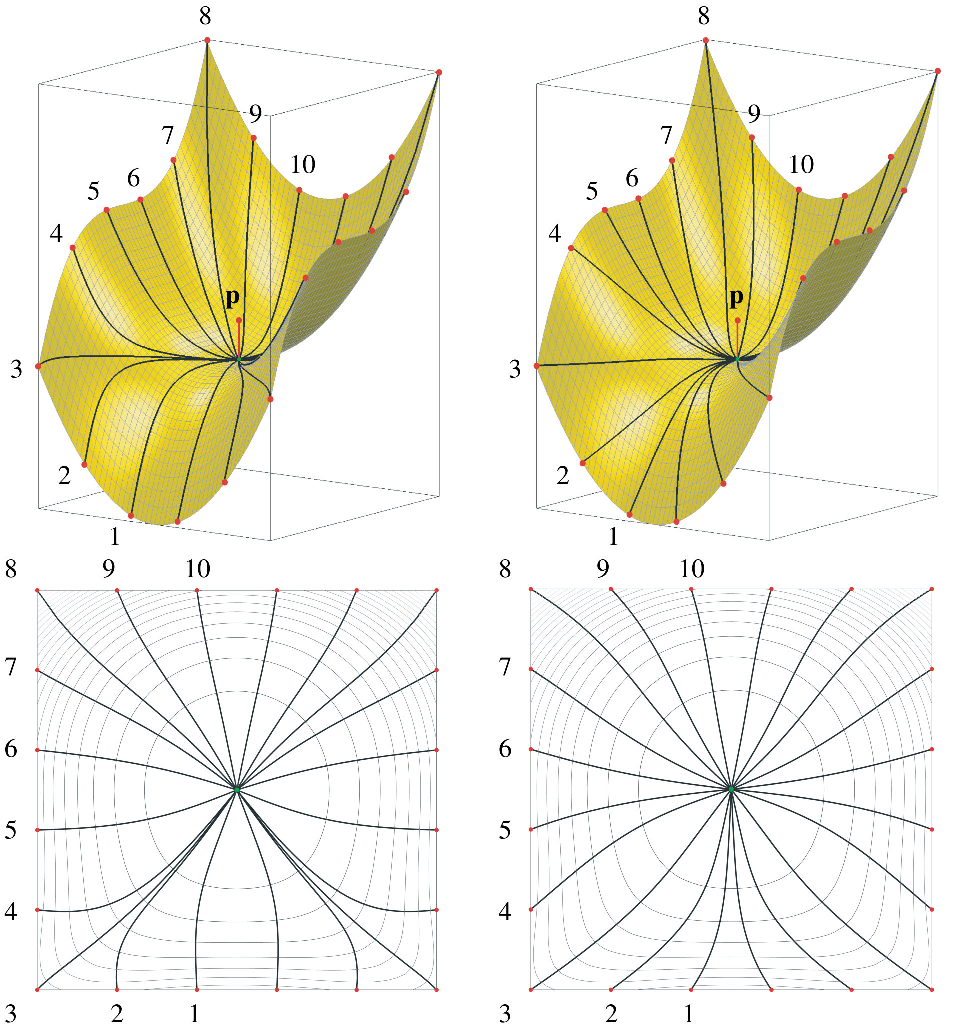


FIGURE 1.6: Gradient flow with respect to the canonical (top-left) and induced (top-right) gradient fields. Below: the corresponding paths on the parameter domain's level sets. The courtesy of the figures goes to Georg Nawratil.

Eq. 1.32 into

$$u_+ = u_c - \gamma \mathbf{g}_{u_c}^{-1}.\nabla f(u_c) = u_c - \gamma.\nabla_g f(u_c). \tag{1.33}$$

Fig. 1.6 shows the gradient flow with respect to the ambient metric (ordinary Euclidean metric) and the induced metric on the surface. The corresponding level sets show the convergence to the final solution p^*.

Computation of Pedal Points

In later chapters, in several occasions, we are interested in obtaining the *closest point* on a certain surface from a point outside of it. This problem is related to the concept of orthogonal projection, yet despite the fact that Definition 4 gives an intuition toward the nature of it but it does not provide a computational tool. In fact there can be several ways to calculate π for a sample point p. But for reasons that will be cleared out later the *Lagrange multiplier method* is chosen.

Definition 9. *Consider the "Lagrange equation" $L := f + \lambda_1 \Phi_1 + ... + \lambda_n \Phi_n$, where f is the smooth "distance function" on a manifold N. Moreover a certain subset S of N is given as*

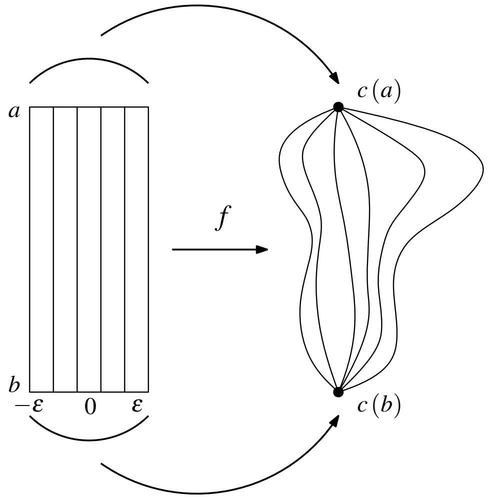


FIGURE 1.7: Illustration of a *proper* variation.

the zero-set of the polynomials $\Phi_1, \ldots, \Phi_n$ and their corresponding "Lagrange multipliers" are denoted by $\lambda_1, \ldots, \lambda_n$. Then the solutions of the system of equations

$$\nabla L = 0 \quad and \quad \Phi_i = 0 \quad for \quad i = 1, ..., n \tag{1.34}$$

are called "pedal points", as these points of S cause local extrema of the distance function.

Note that, the set $\pi(p)$ is a subset of the set S of *pedal points* (see Definition 4).

1.1.9 Curvature of the Curves

The second fundamental topic of interest in differential geometry is the *curvature*. Informally, the curvature is the amount by which a smooth curve deviates from a line, or a surface deviates from being a plane. There are extensive literature on this topic that generalize this notion to higher dimensions and abstract manifolds [O'Neill, 1983]. However, in this thesis only a glance of this concept, namely the *curvature* of the smooth curves, is needed and hence only a brief introduction to this concept is given.

Let $c : (a,b) \longrightarrow \mathbb{R}^n$ be a curve parametrized by *arc length* s (see [Do Carmo, 2016, chapter 1]). Since the tangent vector $c'(s)$ has unit length, the norm $\|c''\|$ of the second derivative measures the rate of change of the angle which neighbouring tangents make with the tangent at s. The curvature $\kappa(s) := \|c''\|$ gives, therefore, a measure of how rapidly the curve pulls away from the tangent line at s, in the neighbourhood of s [Do Carmo, 2016, page 16].

In the later chapters, due to involvement of the piecewise smooth curves, we are mainly interested in the discrete counterparts of the smooth notion such as curvature. Consequently, such an introduction is necessary:

Definition 10 (Milnor, 1950)**.** *In the discrete case, if $\{a' = p_1, p_2, \ldots, p_n = b'\}$ is a sequence of breakpoints defining a piecewise smooth curve, it is natural to consider the angle θ_i between vectors $p_{i+1} - p_i$ and $p_i - p_{i-1}$ as the "curvature" at the breakpoint p_i.*

1.1.10 Energy of Curves

In this section, the variations and energy of a curve, namely the *bending energy* and *geodesic energy*, are introduced. In later chapters, the tools studied here are borrowed to manipulate the behaviour of certain curves *globally*.

Definition 11 (Carmo, 1992, page 192)**.** *Let $c : [a,b] \longrightarrow N$ be piecewise smooth curve on a Riemannian manifold N. A "variation" of c is a continuous mapping $f : (-\varepsilon,\varepsilon) \times [a,b] \longrightarrow N$ such that*

- *$f(0,t) = c(t)$, $t \in [a,b]$,*
- *there exists a subdivision of $[a,b]$ by points $a = t_1 < ... < t_{n-1} < t_n = b$ in a way that the restriction of f to each $(-\varepsilon, \varepsilon) \times [t_i,t_{i+1}]$ is differentiable.*

A variation is said to be "proper" if $f(s,a) = c(a)$ and $f(s,b) = c(b)$ for all $s \in (-\varepsilon,\varepsilon)$ (see Fig. 1.7).

Considering a smooth variation (smooth f), the *geodesic energy E* and *bending energy B* of the variation f for $s \in (-\varepsilon,\varepsilon)$ are defined as follows [Carmo, 1992, page 192, Pottmann and Wallner, 2009, page 298]:

$$E(s) = \int_a^b \| \frac{\partial f}{\partial t}(s,t) \|^2 \, dt, \tag{1.35}$$

$$B(s) = \int_a^b \| \frac{\partial^2 f}{\partial t^2}(s,t) \|^2 \, dt. \tag{1.36}$$

In later chapters, it is shown that a variation of a curve is discretely done and depends on a predefined *step size* (also known as *learning rate*). Hence, an introduction of Eqs. 1.36 with a discrete setting is necessary.

Discretization of Curve Energies

For the sake of simplicity of notations, we restrict ourselves to the case of the piecewise smooth curve in $\mathbb{R}^n$ as they are needed later on.

Definition 12. *Assume that $c : [a,b] \longrightarrow \mathbb{R}^n$ is a "piecewise smooth" curve with a finite partition, $a = t_1 < ... < t_{n-1} < t_n = b$, then the "discrete geodesic energy" E and "discrete bending energy" B of the curve c are as follows (cf. [Pottmann and Wallner, 2009]):*

$$E(c) = \sum_{i=2}^{n} \|c(t_i) - c(t_{i-1})\|^2, \tag{1.37}$$

$$B(c) = \sum_{i=2}^{n-1} \|(c(t_{i+1}) - c(t_i)) - (c(t_i) - c(t_{i-1}))\|^2, \tag{1.38}$$

while "discrete length" and "discrete total curvature" are respectively given by (cf. [Carmo, 1992; Milnor, 1950]):

$$L(c) = \sum_{i=2}^{n} \|c(t_i) - c(t_{i-1})\|, \tag{1.39}$$

$$\tau(c) = \sum_{i=2}^{n-1} \|(c(t_{i+1}) - c(t_i)) - (c(t_i) - c(t_{i-1}))\|. \tag{1.40}$$

Finally, it is noteworthy to emphasize that the variations under study in this paper are *proper* as we require the motions to remain between a fixed start- and end-pose.

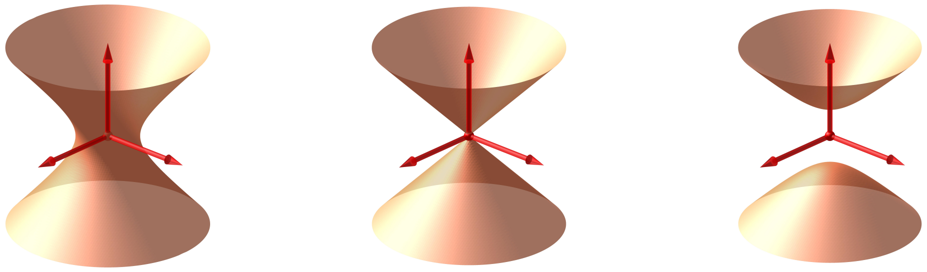


FIGURE 1.8: Illustrations of affine varieties. Left: a one-sheet hyperboloid $\mathbf{V}\left(x^2+y^2-z^2-1\right)$ as a smooth connected surface. Middle: a cone $\mathbf{V}\left(x^2+y^2-z^2\right)$ as a non-smooth connected surface. Right: a two-sheet hyperboloid $\mathbf{V}\left(x^2+y^2-z^2+1\right)$ as a smooth non-connected surface.

1.2 Algebraic Geometry

Algebraic geometry is a branch of mathematics, classically studying zeros of multivariable polynomials. The fundamental objects of study in algebraic geometry are *algebraic varieties*, which are in fact the set of solutions of the system of polynomials. Algebraic geometry occupies a central place in modern mathematics and has multiple conceptual connections with such diverse fields as complex analysis and topology. In the 20th century, algebraic geometry is divided into several subareas, one of which is *computational algebraic geometry*. Computational algebraic geometry is an area that has emerged at the intersection of the algebraic geometry and computer algebra, with the rise of computers. It consists mainly of algorithm design and software development for the study of properties of explicitly given algebraic varieties. Some of the crucial recent applications of this branch revolve around the computer science fields such as robotics and artificial intelligence [Cox, Little, and O'Shea, 2013].
The mathematical discipline of this thesis is closely related to one of the branches of computational algebraic geometry known as *computational kinematics*. This branch which, formally, is defined as the kinematics involving intensive computations [Angeles, 2002], heavily resorts to the algebraic geometry and due to its repeated use through out the thesis, dedicating a subsection to it is unavoidable.
In this subsection, some definitions and theorems are recalled from algebraic geometry. Once again, the presented examples in this subsection are in $\mathbb{R}^3$ so that the reader obtains a geometric intuition for the future cases to come (which are mainly in $\mathbb{R}^6$).

1.2.1 Algebraic Varieties

The main goal of study in the classical algebraic geometry revolves around the geometrical objects known as *algebraic varieties*. The following definition deals with the concept of *affine algebraic varieties*.

Definition 13 (Cox, Little, and O'Shea, 2013, page 5)**.** *Let $\mathbb{K}$ be a field, and let $f_1, \dots f_s$ be polynomials in $\mathbb{K}[x_1,\dots,x_n]$. Then an "affine variety" defined by $f_1,\dots,f_s$ is:*

$$\mathbf{V}(f_1,\dots,f_s) := \{(a_1,\dots,a_n)\in\mathbb{K}^n \mid f_i(a_1,\dots,a_n)=0, \forall i,\ 1\le i\le s\}. \tag{1.41}$$

Thus, an affine variety is the set of all solutions of the system of equations in the space $\mathbb{K}^n$. The reader can find some examples of affine algebraic varieties in Fig. 1.8. Affine algebraic varieties obey certain properties out of which the intersection and union properties are most important.

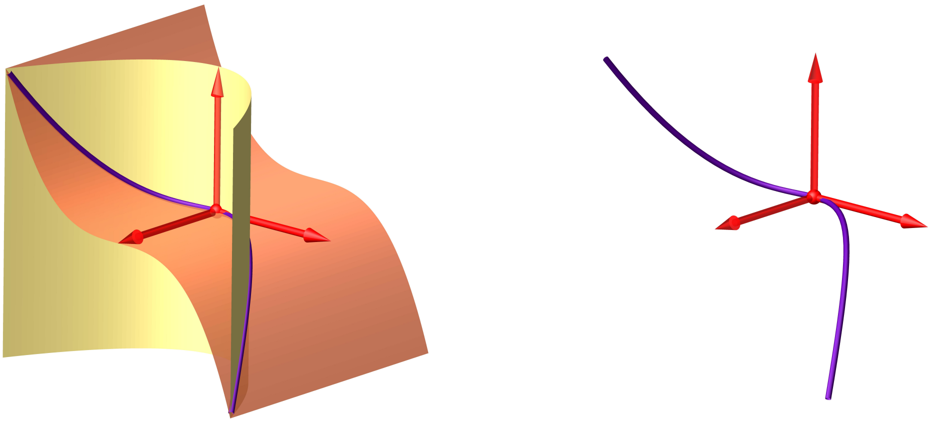


FIGURE 1.9: Illustrations of the twisted cubic. Left: The twisted cubic as the intersection $\mathbf{V}\left(y-x^2\right)\cap\mathbf{V}\left(z-x^3\right)$. Right: A general view.

Lemma 3 (Hartshorne, 2013, page 2)**.** *If $V=\mathbf{V}(f_1,\ldots,f_s)$, $W=\mathbf{V}(g_1,\ldots,g_t)$ are affine varieties, then we have*

$$\mathbf{V}(1)=\emptyset, \tag{1.42}$$

$$\mathbf{V}(0)=\mathbb{K}^n, \tag{1.43}$$

$$V\cap W=\mathbf{V}(f_1,\ldots,f_s,g_1,\ldots,g_t), \tag{1.44}$$

$$V\cup W=\mathbf{V}(f_i\,g_j \mid 1\leq i\leq s,\, 1\leq j\leq t). \tag{1.45}$$

By considering the compliment of the algebraic varieties, the properties listed in Lemma 3 define a topology on $\mathbb{K}^n$ called *Zariski topology*.
Depending on the ambient space, special names are reserved for the varieties that are defined by a single irreducible polynomial. The varieties obtained in this way possess some computational benefits that may be of use in coding algorithms. In later chapters we resort to these varieties from time to time to enjoy the aforementioned advantages.

Definition 14 (Hartshorne, 2013, page 4)**.** *Let f be an irreducible polynomial in $R=\mathbb{K}[x_1,...,x_n]$. Then the affine variety $\mathbf{V}(f)$ is:*

- *an affine curve if $n=2$,*
- *a surface if $n=3$,*
- *a hypersurface if $n>3$.*

By having the basic objects defined, we move to a fundamental theorem in classical algebraic geometry. Hilbert's Nullstelllensatz is a fundamental theorem that establishes relation between geometry and algebra. More specifically, it relates algebraic varieties to ideals in polynomial rings over algebraically closed fields.

Theorem 1 (Cox, Little, and O'Shea, 2013, page 183)**.** *Let $\mathbb{K}$ be an algebraically closed field. If I is an ideal in $R:=\mathbb{K}[x_1,\ldots,x_n]$, then*

$$\mathbf{I}(\mathbf{V}(I))=\sqrt{I}, \tag{1.46}$$

where lhs of Eq. 1.46 stands as the set of all polynomials in R vanishing over $\mathbf{V}(I)$ and the rhs is

$$\sqrt{I} = \{f \in R \mid \exists n \in \mathbb{N}, f^n \in I\}. \tag{1.47}$$

Finally, before shifting to the next class of interesting varieties, an example is given.

Example 9. *The variety $\mathbf{V}\left(y-x^2, z-x^3\right) \in \mathbb{R}^3$ is called the “twisted cubic" (cf. Fig 1.9). This curve has some interesting features such as being “non-planar", “non-symmetric" and "smooth" which makes it a suitable testing ground for rigid body motions in the later chapters.*

1.2.2 Projective Varieties

Metaphorically, from an applied point of view, a projective space is the way that “human eye sees the world", or in another word, is the space with perspectives. Recently, the computational aspects of projective geometry has played different roles in computer science, kinematics, optics and even art.
Here in this section we give a short introduction to the definition of a projective space and projective varieties.
We define an equivalence relation $\sim$ on $\mathbb{K}^{n+1} \setminus \{0\}$ by setting

$$(y_0, \ldots, y_n) \sim (x_0, \ldots, x_n) \tag{1.48}$$

if $\exists \lambda \, (\neq 0) \in \mathbb{K} \setminus \{0\}$ such that $(y_0, \ldots, y_n) = \lambda \, (x_0, \ldots, x_n)$.

Definition 15 (Cox, Little, and O’Shea, 2013, page 368)**.** *The “n-dimensional projective space over the field $\mathbb{K}$", denoted by $\mathbb{P}\mathbb{K}^n$ is the set of equivalence classes of $\sim$ on $\mathbb{K}^{n+1} \setminus \{0\}$. Thus*

$$\mathbb{P}^n = \left(\mathbb{K}^{n+1} \setminus \{0\}\right) / \sim . \tag{1.49}$$

In this way, the n-tuples $(x_0, \ldots, x_n)$ define a point in $\mathbb{P}\mathbb{K}^n$ and are called homogeneous coordinates. These coordinates from now on are denoted by $(x_0 : \ldots : x_n)$.

In projective space the well-defined geometric objects to study are the *projective varieties* which are defined as follows.

Definition 16 (Cox, Little, and O’Shea, 2013, page 370)**.** *Let $\mathbb{K}$ be a field and let $f_1, \ldots, f_s \in \mathbb{K}[x_0, \ldots, x_n]$ be homogeneous polynomials. Then*

$$\mathbf{V}(f_1, \ldots, f_s) := \{(a_0, \ldots, a_n) \in \mathbb{P}^n \mid f_i(a_0, \ldots, a_n) = 0, \forall i \, 1 \leq i \leq s\}, \tag{1.50}$$

is called a projective variety.

Finally, before moving to an example, the following theorem is necessary to be mentioned.

Corollary 1 (Cox, Little, and O’Shea, 2013, page 370)**.** *For each $i = 0, \ldots, n$, let*

$$U_i = \{(x_0, \ldots, x_n) \in \mathbb{P}^n \mid x_i \neq 0\}. \tag{1.51}$$

a) The points of each U_i are in one to one correspondence with the points of $\mathbb{K}^n$,

b) The compliment $\mathbb{P}^n \setminus U_i$ can be identified as $\mathbb{P}^{n-1}$,

c) $\mathbb{P}^n = \bigcup_{i=1}^n U_i$.

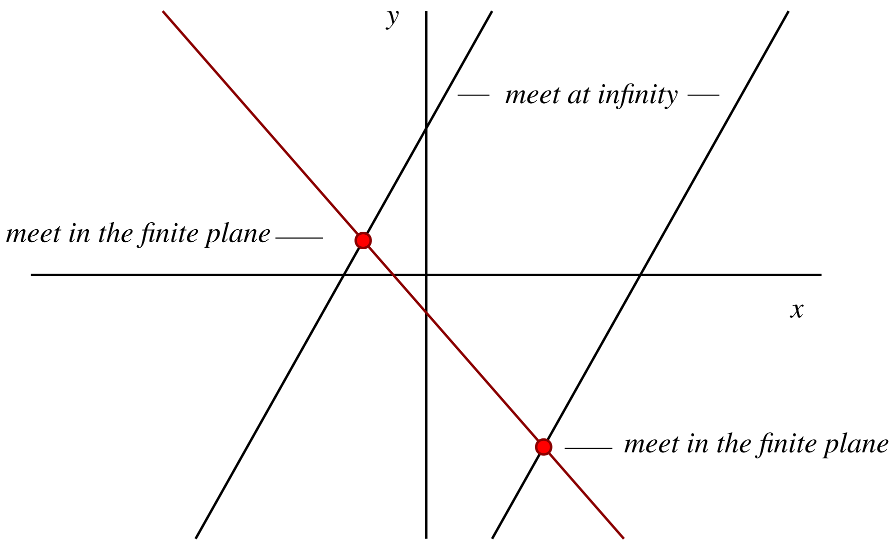


FIGURE 1.10: Illustration of the intersection of lines in the projective plane. The black lines meet on the line at infinity while the red and the blacks meet in the finite plane $\mathbb{R}^2$.

Corollary 1 allows one to think of the projective space as the disjoint union of a *finite space* and a *plane at infinity* (which is in fact a copy of a one dimensional lower projective space). As an example the 3-dimensional real projective space in this sense can be written as $\mathbb{P}^3 = \mathbb{R}^3 \cup \Omega_\infty$, where $\Omega_\infty \cong \mathbb{P}^2$.

Example 10. *Fig. 1.10 shows the case of the intersection of lines in the real projective plane $\mathbb{P}^2$. The parallel lines meet on the line at infinity, like the train rails which meet on the horizon. While the non-parallel ones meet in $\mathbb{R}^2$.*

1.2.3 Critical Values Vs. Singular Points

Due to the nature of the definition of manipulator's singularities, the idea of *singular points* and *critical values* are sometimes mixed up. In order to resolve this, let's visit the definition of *regular points* and *regular values*.

Definition 17. *If $F : M \longrightarrow N$ is a smooth map, a point $p \in M$ is said to be a "regular point" of F if $dF_p : T_pM \longrightarrow T_{F(p)}N$ is surjective; it is a "critical point" otherwise. Note that if $\dim M < \dim N$ then every point of M is going to be a critical point. Furthermore, a point $c \in N$ is a "regular value" if every point of the level set $F^{-1}(c)$ is a regular point and a "critical value" otherwise [Lee, 2003].*

Example 11. *Consider the following smooth surjective map*

$$f : \mathbb{R}^2 \longrightarrow \mathbb{R}, \tag{1.52}$$

$$(x, y) \longmapsto x^2 + y^2, \tag{1.53}$$

computing its derivative gives

$$df = \begin{pmatrix} 2x & 2y \end{pmatrix} \tag{1.54}$$

Evidently, df is not onto at the origin and hence $(0,0)$ will be a critical value of the map f. Geometrically, the critical point is a point on the paraboloid $\mathrm{graph}(f)$ where the tangent space at paraboloid is parallel to one of the coordinate planes ($x - y$ plane in this case).

Now, let's review the definition of *singular points* on an algebraic variety.

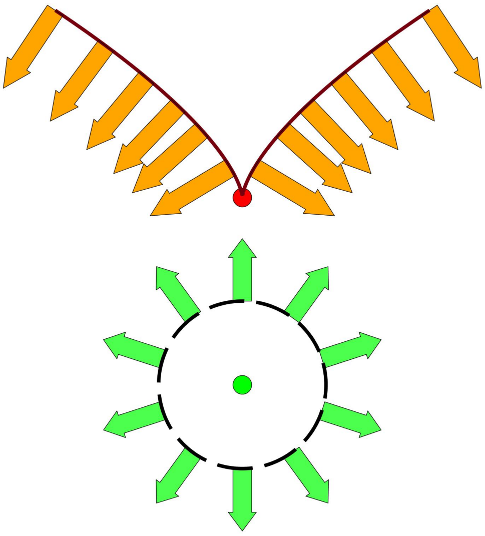


FIGURE 1.11: Illustration of distance of a point in space (green) to a curve with cusp singularity.

Definition 18. *Let p be a point on an affine variety V. Then p is "nonsingular" (or smooth) provided $\dim_p V = \dim T_p V$. Otherwise, p is a "singular point" [Cox, Little, and O'Shea, 2013].*

Theorem 2. *Let $V = \mathbf{V}(f_1, \ldots, f_r) \subset \mathbb{C}^n$ be an arbitrary variety and suppose that $p \in V$ is a point where $J_p(f_1, ..., f_r)$ has rank r. Then p is a nonsingular point of V and lies on a unique irreducible component of V of dimension $n - r$ [Cox, Little, and O'Shea, 2013].*

Example 12. *In order to compute the singular points of a paraboloid consider the following surjective smooth map defining it:*

$$g : \mathbb{R}^3 \longrightarrow \mathbb{R}, \tag{1.55}$$

$$(x, y, z) \longmapsto x^2 + y^2 - z, \tag{1.56}$$

computing it derivative gives

$$dg = \begin{pmatrix} 2x & 2y & -1 \end{pmatrix} \tag{1.57}$$

one easily notes that $\nabla g = dg$. As can be seen dg does not vanish on any point of $\mathbb{R}^3$ including the points on $g^{-1}(0) = \mathbf{V}(x^2 + y^2 - z)$. Hence $(0,0)$ is a critical point for the map f but is not a singular point. Note that, $(0,0)$ is not a critical point for map g.

1.2.4 Distance to a Variety via Lagrange Multipliers

The method of *Lagrange multipliers* is a strategy for finding the local maxima, minima and saddle points of a function f subject to equality constraints g_i (i.e. subject to the condition that one or more equations have to be satisfied exactly by the chosen values of the variables). Mathematically, this method states that $\nabla f = \sum_i \lambda_i \nabla g_i$ at a critical point.

However, in problems to come we mainly deal with the case where the function f is a distance function (from a point to a surface) and the constraint g is a polynomial describing an algebraic variety. Geometrically, this means that a critical point is obtained whenever the gradient fields on the variety and the on the shape defined by the metric become co-linear at a point. An important consequence of this interpretation is that the *Lagrange multipliers method* fails to identify critical points at the singularities of a variety. Note that this is expected as the gradient vector vanishes at the singular points of a variety.

Example 13. *Consider the variety* $V = \mathbf{V}\left(x^2 - y^3\right) \subset \mathbb{R}^2$ *and a point outside of it* $A = (0, -1)$ *(cf. Fig. 1.11). The Lagrange multiplier system for this case will be*

$$\mathscr{L} = x^2 + (y+1)^2 + \lambda\, \left(x^2 - y^3\right), \tag{1.58}$$

which upon solving yields complex solutions. In another word the closest point of the cusp to the red disk, which is it sharp point, is not detected by the "Lagrange multipliers technique".

Despite this failure of *Lagrange multipliers*, this method will be of great help in forthcoming chapters. The singular points of the variety can be identified separately. In cases to come in future these singular points either form point clouds (finite points in space) or well-behaved geometrical objects such as low dimensional planes and hence can be treated separately.

1.2.5 Rational Varieties

Rational varieties form another interesting class of algebraic varieties which are formally defined as follows.

Definition 19 (Cox, Little, and O'Shea, 2013, page 252)**.** *Let* $\mathbb{K}$ *be a field and* $V \subset \mathbb{K}^m$ *and* $W \subset \mathbb{K}^n$ *be irreducible affine varieties. A* rational mapping *from* V *to* W *is a function* ϕ *represented by*

$$\phi : V \dashrightarrow W \quad \textit{with} \quad \phi(x_1, \ldots, x_m) = \left(\frac{f_1(x_1, \ldots, x_m)}{g_1(x_1, \ldots, x_m)}, \ldots, \frac{f_n(x_1, \ldots, x_m)}{g_n(x_1, \ldots, x_m)}\right) \tag{1.59}$$

where $\frac{f_i}{g_i} \in \mathbb{K}(x_1, \ldots, x_m)$ *and satisfies the following properties:*

1. ϕ *is defined at some point of* V.
2. *For every* $(a_1, \ldots, a_m) \in V$ *where* ϕ *is defined,* $\phi(a_1, \ldots, a_m) \in W$.

Definition 20 (Cox, Little, and O'Shea, 2013, page 253)**.** *Two irreducible varieties* V *and* W *are said to be* birationally equivalent *if there exist rational mappings* $\phi : V \dashrightarrow W$ *and* $\psi : W \dashrightarrow V$ *such that* $\phi \circ \psi$ *and* $\psi \circ \phi$ *be equal to* id_W *and* id_V *respectively.*

Definition 21 (Cox, Little, and O'Shea, 2013, page 254)**.** *A* rational variety *is a variety that is* birationally *equivalent to* $\mathbb{K}^n$.

One can find the extensive discussion of above definitions in [Shafarevich and Reid, 1994].
Having a rational parametrization of a variety has numerous advantages: If the coefficients of the polynomials f_i and g_i of Eq. (1.59) belong to $\mathbb{Q}$ and if $(x_1, \ldots, x_m)$ is an element of $\mathbb{Q}^m$, then one obtains points with rational coordinates on the singularity variety [Shafarevich and Reid, 1994, page 8]. This is a matter, which is of high importance to computer aided designs, as computers can calculate rational coordinates at a much faster rate. Moreover the rationality of the variety implies that it is *path connected*, which means that every point can be connected to any other point by a continuous curve [Husty and Gosselin, 2008]. This property, in the kinematics of manipulators, is of importance as it can be used for a computationally efficient approximation of the singularity-free workspace by hierarchical structured hyperboxes, where only their boundaries have to be checked to be free of singularities.
The following examples illustrate two rational parametrizations for the case of hyperboloid of one-sheet.

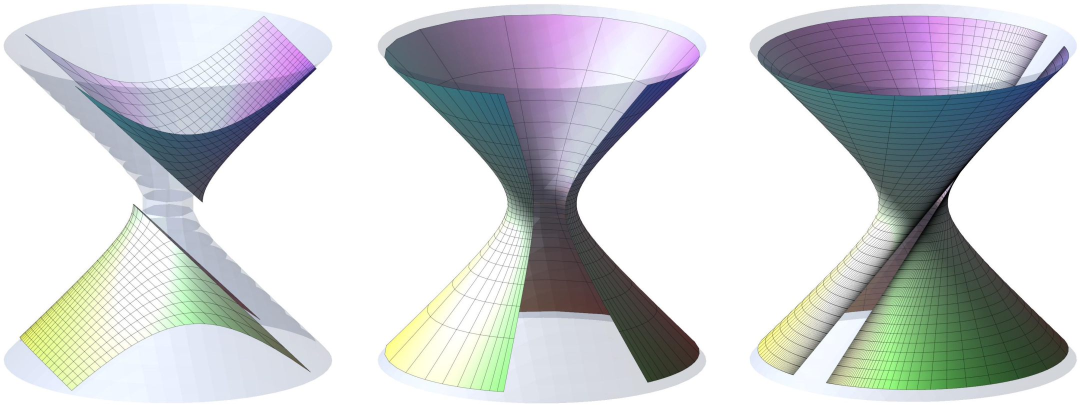


FIGURE 1.12: Illustrations of the comparison of three rational parametrizations of a one-sheet hyperboloid. Left: Rational parametrization using a bundle of line with apex on the plane at infinity. Middle: Rational parametrization using Weierstrass substitution. Right: Rational parametrization of the one-sheet hyperboloid as a ruled surface. The blue silhouette is the hyperboloid that is to be covered by the coloured parametrized surfaces.

Example 14. *The goal is to find a rational parametrization for the one-sheet hyperboloid with the following implicit equation*

$$P := \frac{x^2}{A^2} + \frac{y^2}{B^2} - \frac{z^2}{C^2} - 1. \tag{1.60}$$

One standard way to obtain rational parametrization for quadric surfaces is to resort to the projective space [Schicho, 1998]. Homogenizing the polynomial P from Eq. 1.60 gives a projective variety in $\mathbb{P}^3$ represented by the zero of the following polynomial:

$$P_h := \frac{x^2}{A^2} + \frac{y^2}{B^2} - \frac{z^2}{C^2} - w^2. \tag{1.61}$$

For our purpose we need to compute a birational map from $\mathbf{V}(P_h)$ to $\mathbb{P}^2$. This is easy as if the point $q \in \mathbf{V}(P_h)$ is a smooth point then the projection from q to $\mathbb{P}^2$ is birational. This is equivalent to choosing three linearly independent linear forms vanishing at a smooth point, here namely $q = (0 : B : C : 0)$. For details on smoothness of algebraic varieties see [Cox, Little, and O'Shea, 2013, page 490] and for details on the general parametrization algorithm for the quadric surfaces see [Schicho, 1998]. Then we have the following map:

$$\phi : U \subset \mathbf{V}(P_h) \dashrightarrow \mathbb{P}^2, \tag{1.62}$$

$$(x : y : z : w) \mapsto (x : Cy - Bz : w), \tag{1.63}$$

where $U = \mathbf{V}(P_h) \setminus q$. This gives a rational map from our quadric minus a point to $\mathbb{P}^2$. Geometrically, this is similar to the idea of parametrizing a circle using a pencil of lines. Now, name

$$X = x, \tag{1.64}$$

$$Y = Cy - Bz, \tag{1.65}$$

$$Z = w, \tag{1.66}$$

and solving the above equations for x, y, and w and substituting the result in Eq. 1.61, one obtains z. Finally, the rational parametrization is:

$$\psi : U' \subset \mathbb{P}^2 \dashrightarrow \mathbf{V}(P_h)$$
$$(X:Y:Z) \mapsto \left(X : \frac{(-C^2B^2+A^2)\,Y}{2A^2C} + \frac{B^2CZ^2}{2Y} : \frac{(-C^2B^2-A^2)\,Y}{2A^2B} + \frac{BC^2Z^2}{2Y} : Z \right)$$

where $U' = \mathbb{P}^2 \setminus \mathbf{V}(Y)$. The rest is easy to check that ψ and ϕ give the birational equivalency of $\mathbf{V}(P_h)$ and $\mathbb{P}^2$. By putting $Z = 1$, one achieves the rational parametrization for hyperboloid of one-sheet.
Finally, it could be helpful to take a look at the visual result of our parametrization. Fig. 1.12 - left provides the rational parameterization picture for $A = \sqrt{b}$, $B = \sqrt{4b}$ and $C = \sqrt{4b}$ when $b = 0.5$. The blue hologram of hyperboloid is created using its implicit function while the "more accurate" portions of it are produced using our rational parametrization for $X \in [-2,2]$ and $Y \in [-18,-0.8] \cup [0.8,18]$. Playing with the ranges of X and Y reveals that the parametric portion does not grow identically in all directions and though this rational parametrization gives some theoretical and computational benefits, its not always the best parametric representation for a 3-dimensional visualization of a surface.

In this next example, we repeat the same goal as of Example 14 using a tool of different nature which turns out to be more compatible with plottings.

Example 15. *One useful tool to obtain the rational parametrization of the quadrics is the so-called "Weierstrass substitution", $t \mapsto 2\arctan u$. Combining it with the usual trigonometric relations*

$$\cos^2 t + \sin^2 t = 1 \longrightarrow \left(\frac{1-u^2}{1+u^2}\right)^2 + \left(\frac{2u}{1+u^2}\right)^2 = 1$$

$$\sec^2 t - \tan^2 t = 1 \longrightarrow \left(\frac{1+u^2}{1-u^2}\right)^2 - \left(\frac{2u}{1-u^2}\right)^2 = 1$$

builds up the rational parametrizations for the quadric surfaces.
For the specific case of the one-sheet hyperboloid of Eq. 1.60, the trigonometric parametrization is:

$$x = A\cos\theta\sec\lambda, \tag{1.67}$$
$$y = B\sin\theta\sec\lambda, \tag{1.68}$$
$$z = C\tan\lambda. \tag{1.69}$$

Using the "Weierstrass substitution" yields

$$x = A\frac{(1-u^2)\,(1+v^2)}{(1+u^2)\,(1-v^2)}, \tag{1.70}$$
$$y = B\frac{2u\,(1+v^2)}{(1+u^2)\,(1-v^2)}, \tag{1.71}$$
$$z = C\frac{2v}{1-v^2}. \tag{1.72}$$

One observes the result of such parametrization in Fig. 1.12 - right.

Example 16. *Another well known parametrization of the one-sheet hyperboloid is the one obtained by resorting to the fact that this variety is a ruled surface ((for details on ruled* *surfaces see [Carmo, 1992])). Computing the tangent space to this variety at each point and intersecting the result with it yields two lines. Each of these lines can be utilized to generate the surface. The following rational parametrization describes the one-sheet hyperboloid by sweeping it with the generating line:*

$$(u,v) \mapsto \begin{pmatrix} -A\,(A^2u^2-B^2)\,/\,(A^2u^2+B^2) \\ (2uAB^2)\,/\,(A^2u^2+B^2) \\ 0 \end{pmatrix} + v \begin{pmatrix} -2\,(A^2Bu)\,/\,(A^2u^2+B^2) \\ -B\,(A^2u^2-B^2)\,/\,(A^2u^2+B^2) \\ C \end{pmatrix},$$

where u and v each sweep the real line $\mathbb{R}$*. Fig. 1.12 - right depicts this parametrization. Note that the generating line is visible as one of the two coordinate curves.*

1.2.6 Real Algebraic Geometry

The real algebraic Geometry is a branch of algebraic geometry focusing on the *real* solutions to the algebraic equations and the related mappings between them (i.e. real polynomial mappings). The main geometrical objects of study in this field are the *semialgebraic sets*. Informally, these objects illustrate the real solutions to algebraic inequalities with real number coefficients.

Definition 22. *A* ***semialgebraic*** *set is a subset* S *of* $\mathbb{K}^n$ *for some real closed field* $\mathbb{K}$ *(for example* $\mathbb{K}$ *could be the field of real numbers) defined by a finite sequence of polynomial equations of the form* $P(x_1,...,x_n) = 0$ *and inequalities of the form* $Q(x_1,...,x_n) > 0$*, or any finite union of such sets.*

Similarly to algebraic subvarieties, finite union and intersections of semialgebraic sets are still semialgebraic sets. Furthermore, unlike subvarieties, the complement of a semialgebraic set is again semialgebraic.

The main goal of referring to real algebraic geometry in this thesis is related to use a specific tool known as *cylindrical algebraic decomposition*. In later chapters, the application of this tool in obtaining the workspace of the linear pentapods is depicted.

Definition 23. *Given a set* S *of polynomials in* $\mathbb{R}[x_1,\ldots,x_n]$*, a* ***cylindrical algebraic decomposition*** *is a decomposition of* $\mathbb{R}^n$ *into connected semialgebraic sets called* ***cells****, on which each polynomial has constant sign (i.e. either* $+$*,* $-$ *or zero). To be* ***cylindrical****, this decomposition must satisfy the following condition:*

- *If* $1 \leq i \leq n$ *and* π *is the projection from* $\mathbb{R}^n$ *to* $\mathbb{R}^{n-k}$ *by removing the last k coordinates, then for every pair of cells C and D, one has either* $\pi(C) = \pi(D)$ *or* $\pi(C) \cap \pi(D) = \emptyset$*.*

1.2.7 Singular Value Decomposition

Informally, the *singular value decomposition* (SVD) is a factorization of real or complex matrices. Formally, the singular value decomposition of an $m \times n$ real or complex matrix M is a factorization of the form USV^*, where U is an $m \times m$ real or complex unitary matrix, S is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V is an $n \times n$ real or complex unitary matrix. The diagonal entries σ_i of S are known as *singular values* of M. The SVD is implemented in a vast variety of software libraries including MATLAB and Maple.

Computationally, one of the main usages of the SVD is in obtaining the structure of the null space of a linear map/matrix.

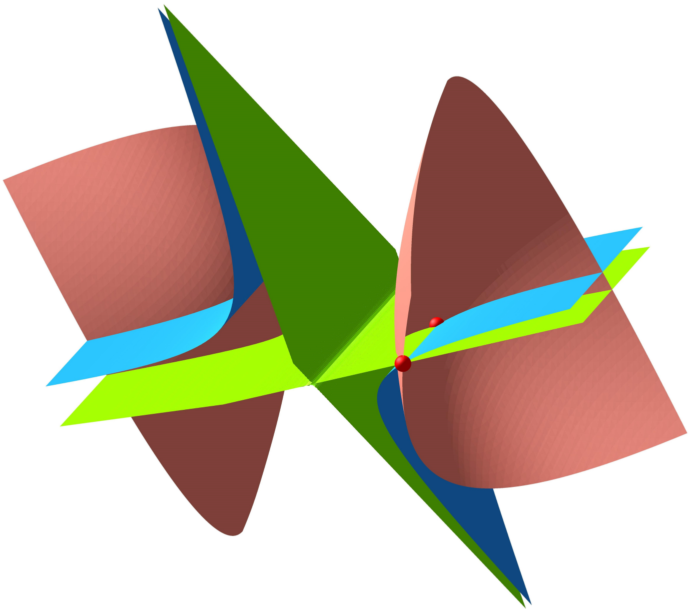


FIGURE 1.13: Illustration of the intersection of three algebraic varieties (given in blue, green and salmon) in two points (red spheres). The courtesy of the related polynomials goes to Franz Winkler.

Example 17. *Assume that W is an inner product space and V is an n-dimensional subspace of it. Then the space of all orthogonal vectors to V is defined by*

$$V^{\perp} := \{w \in W \mid \langle v, w \rangle = 0, \forall v \in V\}, \tag{1.73}$$

[O'Neill, 1983].
Resorting to the matrix format of the inner product yields $v^T g w = 0$ where g is matrix of metric tensor with respect to coordinate axis. Let

$$A := \begin{pmatrix} v_1^T \\ \vdots \\ v_n^T \end{pmatrix}, \tag{1.74}$$

such that $\{v_1, \ldots, v_n\}$ spans V. From basic linear algebra the it is known that $V^{\perp} = null\,(A\,g)$. Using SVD algorithm on the matrix $A\,g$ will result in $U\,S\,V^*$ where the rows of U form the orthonormal basis of V.
In MATLAB this is obtained using the following function:

```
[U, ~, ~] = null(A * g);
```

1.2.8 Resultants method

Informally, the resultant of two polynomials is a polynomial expression which is equal to zero if the polynomials have a common root, or equivalently, a common factor. In this thesis, the resultants are used as a computational tool for solving systems of polynomial equations. The interested reader can find the computational details of these algorithms in [Cox, Little, and O'Shea, 2006, Cox, Little, and O'Shea, 2013].

```
R = resultant(f1, f2, x);
```

1.2.9 Gröbner basis method

The Gröbner basis method allows one to solve problems about polynomial ideals (i.e. polynomial system of equations) in an algorithmic and computational fashion. It is also used in several computational algebra systems (such as Maple) to study specific polynomial ideals that arise in applications [Cox, Little, and O'Shea, 2013, chapter 2].
Suppose J is our polynomial ideal (system of polynomial equations) then two of the useful Maple commands of this method are given below:

```
 with(Groebner);
 with(PolynomialTools);

B := Basis(J, plex(X));
B := Basis(J, lexdeg([X], [u, A]), method = fgb);
```

Example 18. *Suppose that we want to solve the following system of polynomial equations:*

$$f_1 = y^2 - xy - xz + 1, \tag{1.75}$$

$$f_2 = 2xy - 2xz + x^2 + x, \tag{1.76}$$

$$f_3 = 2xy - 2xz + x^2 + 1. \tag{1.77}$$

Each polynomial equation in this system is geometrically interpreted as one of the algebraic varieties given in Fig. 1.13, and solving this system is equivalent to obtaining the intersection of these varieties. Here, our goal is to solve this system via the methods of Gröbner basis and resultants in Maple.

Resultant

Using the command given in Section 1.2.9 we compute:

```
R1 := resultant(f2,f3,z)
R2 := resultant(f1,f3,z)
R3 := resultant(f1,f2,z)
```

which gives

$$R_1 = 2x\,(x-1)\,, \tag{1.78}$$

$$R_2 = -x^3 - 4x^2y + 2xy^2 + x, \tag{1.79}$$

$$R_3 = -x\left(x^2 + 4xy - 2y^2 + x - 2\right), \tag{1.80}$$

```
G1 := resultant(R2, R3, y)
G2 := resultant(R1, R3, y)
G3 := resultant(R1, R2, y)
```

which gives the following equation as the only solution

$$4x^4\,(x-1)^2 = 0, \tag{1.81}$$

$$4x^2\,(x-1)^2 = 0. \tag{1.82}$$

Upon solving these equations and back-substituting it to the rest of the above equation we find that the two points $A = (1,0,1)$ and $B = (1,2,3)$ are the only real solutions to our system of Eq. 1.77.

Gröbner basis

Using the Gröbner basis, we create the ideal $J := \langle f_1, f_2, f_3 \rangle$. Now using one of the mentioned command in Section 1.2.9,

```
B := Basis(J, plex(x,y,z));
```

yields the following basis for the ideal J which is consisted of simpler polynomials compared to the original basis of the aforementioned ideal.

$$\begin{aligned} z^2-4z+3 &= 0, && (1.83)\\ y-z+1 &= 0, && (1.84)\\ x-1 &= 0. && (1.85) \end{aligned}$$

Hence, solving Eq. 1.85 for x and back-substituting it to the rest of the above equation gives the two points $A = (1,0,1)$ and $B = (1,2,3)$ as the only solutions of our system of equations (see Fig. 1.13 for the related geometrical interpretation).

1.2.10 Bézout's Theorem and Extension of Example 18

From time to time in tasks involving computational kinematics one has to estimate the number of solutions of a system of polynomial equations. This is equal to thinking of a zero dimensional variety. The theorem originally states that the number of common points of two curves is at most equal to the product of their degrees, an equality holds if the points at infinity, points with complex coordinates and intersection multiplicities (for definition of multiplicity see [Cox, Little, and O'Shea, 2013, page 139]) are counted [Cox, Little, and O'Shea, 2013, chapter 8]. Bézout's Theorem also refers to higher dimensions which the definition is as follows:

Definition 24 (Cox, Little, and O'Shea, 2006, page 97)**.** *If the equations $F_0 = \ldots F_{n-1} = 0$ have degrees $d_0, \ldots, d_{n-1}$ and finitely many solutions in $\mathbb{P}\mathbb{C}^n$, then the number of solutions (counted with multiplicity) is the multiplication of the aforementioned degrees $d_0 d_1 \ldots d_{n-1}$.*

Geometrically, Definition 24 implies that the number of points obtained after intersecting n hypersurfaces in $\mathbb{P}\mathbb{C}^n$ is $d_0 d_1 \ldots d_{n-1}$ (after counting multiplicity). As soon as we step down from resorting to $\mathbb{P}\mathbb{C}^n$ (or from counting multiplicities) this theorem provides only an upper bound. Chapter 8 of [Cox, Little, and O'Shea, 2013] contains beautiful yet simple example of this theorem for intersecting planar curves. Here we rely on Example 18 to demonstrate the Bézout's theorem.

Example 19. *Consider the equations of Example 18 in the projective space $\mathbb{P}\mathbb{C}^3$ which gives the following three hyperquadrics:*

$$\begin{aligned} f_1 &= y^2 - xy - xz + x_0^2, && (1.86)\\ f_2 &= 2xy - 2xz + x^2 + xx_0, && (1.87)\\ f_3 &= 2xy - 2xz + x^2 + x_0^2, && (1.88) \end{aligned}$$

where x_0 is our extra homogenizing variable. At this point Bézout's theorem states that the number of solutions to this system is 8 (having in mind that the multiplicity is counted as well). Upon solving the above system of polynomial equations one finds the following five *points:*

$$(1:2:3:1)\,, \tag{1.89}$$

$$(1:0:1:1)\,, \tag{1.90}$$

$$\left(-2+\sqrt{6}:1:\sqrt{6}/2:0\right), \tag{1.91}$$

$$\left(-2-\sqrt{6}:1:-\sqrt{6}/2:0\right), \tag{1.92}$$

$$(0:0:1:0)\,, \tag{1.93}$$

out of which, the magenta ones are exactly those which are obtained in Example 18 (in finite space) and the rest are on the plane at infinity (the last coordinates refers to x_0).

Chapter 2

Linear Pentapods

2.1 Introduction

A *(general) linear pentapod* (cf. Fig. 2.1) is defined as a five degree-of-freedom (DoF) *line-body component* of a Gough-Stewart platform consisting of a linear motion platform ℓ with five identical spherical-prismatic-spherical (S$\underline{\text{P}}$S) legs, where the prismatic joints are active and the rest are passive [Kong and Gosselin, 2001]. The pose of ℓ is uniquely determined by a position vector $\mathbf{p} = (u_4, u_5, u_6)^T \in \mathbb{R}^3$ and an orientation given by a unit-vector $\mathbf{i} = (u_1, u_2, u_3)^T \in \mathbb{R}^3$. The coordinate vector $\mathbf{m}_j$ of the platform anchor point m_j of the j-th leg is defined by the equation $\mathbf{m}_j = \mathbf{p} + r_j\mathbf{i}$ with $r_j \in \mathbb{R}$ and the base anchor points M_j of the j-th leg has coordinates $\mathbf{M}_j = (x_j, y_j, z_j)^T$ for $j = 1, \ldots, 5$.

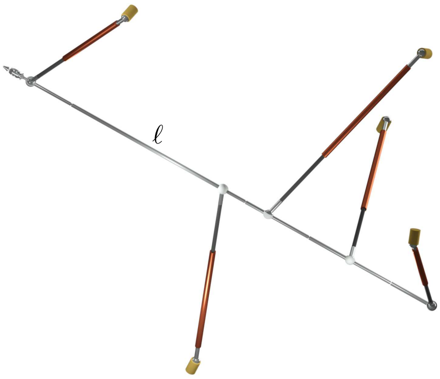


FIGURE 2.1: Sketch of a linear pentapod.

It turns out that this kind of manipulator is an interesting alternative to serial robots handling axis-symmetric tools. Some fundamental industrial tasks such as 5-axis milling, laser engraving and water jet cutting are counted as its applications in industry [Borràs, Thomas, and Torras, 2010b; Weck and Staimer, 2002].

Singularity analysis plays an important role in motion planning of parallel manipulators. Special configurations referred to as *kinematic singularities* have always been central in mechanism theory and robotics. Beside being an intellectually appealing topic, the study of kinematic singularities provides an insight of major practical and theoretical importance for the design, control, and application of robot manipulators.

In such singularities, the *kinetostatic* properties of a mechanism undergo sudden and dramatic changes. This motivates the enormous practical value of a careful study and thorough understanding of the phenomenon for the design and use of manipulators.

2.1.1 Review

Here, a brief review on the previous findings related to the singularity analysis of the linear pentapods is presented.

Forward and Inverse Kinematics of the Linear Pentapods

From a geometrical point of view, the forward kinematics problem of the linear pentapod is equivalent to the problem of placing a rigid body such that five given points of the body lie on five spheres with center $\mathbf{M}_i$ and radius ρ_i [Borràs, Thomas, and Torras, 2010a, page 9]. Geometrically, this is equivalent to finding the solutions of the following system:

$$\rho_i^2 = \|\mathbf{m}_i - \mathbf{M}_i\| = \|\mathbf{p} + r_i\,\mathbf{i} - \mathbf{M}_i\|. \tag{2.1}$$

On the other hand, the inverse kinematics deals with the problem to determine the leg lengths if the end-effector position is given by $\mathbf{p}$ and $\mathbf{i}$. Based on the results in [Bär and Weiß, 2006, Borràs et al., 2009] the forward and inverse kinematics of the linear pentapods are given in the following table:

Forward/Inverse Kinematics' Number of Solutions		
	Forward kinematics	Inverse Kinematics
General linear pentapod	8	1

TABLE 2.1: A summary on the number of solutions of the forward and inverse kinematics of the linear pentapods.

In this context it should be mentioned that the forward kinematics of a linear pentapod was solved for the first time in [Zhang and Song, 1992] under the assumption of a planar base, and in [Nawratil and Schicho, 2017] for the general case.

Redundancy

It is well known that in applications parallel manipulators have some crucial advantages over the serial counterparts (e.g. stiffness in certain directions, higher accuracy, etc). However, one of the fundamental disadvantages of this class of manipulators is their limited workspace. In order to ameliorate these forgoing advantages, redundant parallel manipulators have been designed. Most of the current designs are based on the concept of actuation redundancy in which more actuators than necessary are used for the control of the robot [Dasgupta and Mruthyunjaya, 1998, Kim, 1997, Wu et al., 2009, Merlet, 1996, Nakamura and Ghodoussi, 1989]. In contrast with actuation redundancy, kinematic redundancy involves the introduction of additional degrees of freedom. Additionally, they generally involve more controlled motion parameters or actuators then the minimum required for a generic set of tasks [Wang and Gosselin, 2004]. Examples of such manipulators can be found within [Zanganeh and Angeles, 1994b, Zanganeh and Angeles, 1994a, Huang and Kong, 1995].

Singularity Analysis of Linear Pentapods

If the system presented in Eq. 2.1 is written as an implicit relation between the leg lengths $\rho = (\rho_1, \ldots, \rho_5)$ and the linear pentapod's pose $u = (u_1, \ldots, u_6)$ then the system can be abbreviated as

$$F_i(u, \rho) = 0, \quad \forall i,\ 1 \le i \le 5. \tag{2.2}$$

Differentiating Eqs. 2.2 with respect to time gives

$$\mathbf{J}\dot{u} + \mathbf{K}\dot{\rho} = 0, \tag{2.3}$$

where $\mathbf{J} = \partial F / \partial (u_1, \dots, u_6)$ and $\mathbf{K} = \partial F / \partial (\rho_1, \dots, \rho_5)$. The structure of Eq. 2.3 bears an interesting analysis regarding the singularities of the linear pentapods. Specializing the concepts of *serial singularities* and *parallel singularities* defined in [Merlet, 2006] for the case of linear pentapod gives

Definition 25. *A linear pentapod has "serial singularities" if the Jacobian matrix* **K** *is rank deficient.*

Definition 26. *A linear pentapod has "parallel singularities" if the Jacobian matrix* **J** *is rank deficient.*

Depending on the singularity type, one of the following scenarios happens:

- If the pentapod has *only* the serial singularities then for certain non-zero joint velocities, namely $\dot{\rho}$, the matrix multiplication $\mathbf{K}\dot{\rho}$ vanishes. This makes the pose velocity, $\dot{u}$, to vanish. In another word the manipulator loses degree(s)-of-freedom.

- If the pentapod has *only* parallel singularities then for certain non-zero pose velocities, namely $\dot{u}$, the matrix multiplication $\mathbf{J}\dot{u}$ vanishes, which results in the vanishing of $\dot{\rho}$. In another word, the prismatic joints are locked (producing no velocity) but the motion platform is locally moveable. This can be seen as gaining degree(s)-of-freedom.

- If the pentapod has *both* singularities, then both previous scenarios may occur.

The singularity analysis of linear pentapods has undergone an acceptable level of investigations over the past few years. In the following we give an overview of the obtained results:

From the line-geometric point of view (cf. [Merlet, 1989]) a linear pentapod is in a singular configuration if and only if the five carrier lines of the legs belong to a linear line congruence [Pottmann and Wallner, 2009]; i.e. the Plücker coordinates of these lines are linearly dependent. From this latter characterization the following algebraic one can be obtained (cf. [Rasoulzadeh and Nawratil, 2018]):

There exists a bijection between the configuration space of a linear pentapod and all points $(u_1, u_2, u_3, u_4, u_5, u_6)^T \in \mathbb{R}^6$ located on the singular quadric $\Gamma : u_1^2 + u_2^2 + u_3^2 = 1$, where (u_1, u_2, u_3) determines the orientation of the linear platform ℓ and (u_4, u_5, u_6) its position. Then the set of all singular robot configurations is obtained as the intersection of Γ with a cubic hypersurface Σ of $\mathbb{R}^6$, which can be written as $\Sigma : \det(\mathbf{S}) = 0$ (which from now on will be called *singularity polynomial*) with

$$\mathbf{S} = \begin{pmatrix} 1 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ 0 & u_4 & u_5 & u_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & u_1 & u_2 & u_3 \\ r_2 & x_2 & y_2 & z_2 & r_2x_2 & r_2y_2 & r_2z_2 \\ r_3 & x_3 & y_3 & z_3 & r_3x_3 & r_3y_3 & r_3z_3 \\ r_4 & x_4 & y_4 & z_4 & r_4x_4 & r_4y_4 & r_4z_4 \\ r_5 & x_5 & y_5 & z_5 & r_5x_5 & r_5y_5 & r_5z_5 \end{pmatrix}. \tag{2.4}$$

(according to [Borràs, Thomas, and Torras, 2010b]) under the assumption that $x_1 = y_1 = z_1 = r_1 = 0$. Note that this assumption can be done without loss of generality as the fixed/moving frame can always be chosen in a way that the first base/platform anchor point is its origin.

Remark 5. *The relation between the matrix* **S** *and the Plücker coordinates of the pentapod's legs is not well explained in the literature. For the interested reader this gap is closed in the Appendix* A.

A singular configuration can also be characterized as a multiple solution of the direct kinematics problem. If the direct kinematics problem has a continuous solution, then the linear pentapod has a so-called self-motion. Note that each pose of such a motion is a singular one. All designs of linear pentapods possessing such motions are listed in [Nawratil and Schicho, 2017]. A more detailed study of the corresponding self-motions is performed in [Nawratil, 2018]. Moreover the last two cited papers also contain extensive literature reviews on this topic.

A further well-studied field within the singularity analysis of linear pentapods are designs, which are singular in any configuration. These so-called architecture singular designs are completely classified in [Nawratil and Schicho, 2017, Section 1.3], where also all relevant references in this context are cited.

Finally it should be noted, that Borràs and Thomas have studied how to move the leg attachments in the base and the platform of 5-S$\underline{\text{P}}$S linear pentapod without altering the robot's singularity locus (for a planar base see [Borràs, Thomas, and Torras, 2011b] and for a non-planar one see [Borràs, Thomas, and Torras, 2010b]).

2.1.2 Notation and preparatory work

The following notations are used in the rest of the thesis:

- The compact notations $\mathbf{X} = (x_2, x_3, x_4, x_5)^T$, $\mathbf{Y} = (y_2, y_3, y_4, y_5)^T$, $\mathbf{Z} = (z_2, z_3, z_4, z_5)^T$ are introduced for the coordinates related to base anchor points.
- The compact notation $\mathbf{r} = (r_2, r_3, r_4, r_5)^T$ is used for the coordinates related to platform anchor points.
- The *component-wise* product of two vectors (also known as the *Hadamard product*) is given as follows:

$$\mathbf{r}\circ\mathbf{X} = \begin{pmatrix} r_2x_2 \\ r_3x_3 \\ r_4x_4 \\ r_5x_5 \end{pmatrix}, \quad \mathbf{r}\circ\mathbf{Y} = \begin{pmatrix} r_2y_2 \\ r_3y_3 \\ r_4y_4 \\ r_5y_5 \end{pmatrix}, \quad \mathbf{r}\circ\mathbf{Z} = \begin{pmatrix} r_2z_2 \\ r_3z_3 \\ r_4z_4 \\ r_5z_5 \end{pmatrix}. \tag{2.5}$$

- For the sake of simplicity in notation as well as interpretation, we use the *bracket*; i.e.:

$$[\mathrm{A}_1, \mathrm{A}_2, \mathrm{A}_3, \mathrm{A}_4] = \det(\mathbf{A_1}, \mathbf{A_2}, \mathbf{A_3}, \mathbf{A_4}) \quad \text{with} \quad \mathbf{A_i} \in \{\mathbf{r}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{r}\circ\mathbf{X}, \mathbf{r}\circ\mathbf{Y}, \mathbf{r}\circ\mathbf{Z}\}.$$

  It is noteworthy that in the coming subsections of this chapter, the Roman letters inside the bracket are interpreted as points in projective space while the bold letters denote the corresponding homogeneous coordinates expressed as a vector.

Furthermore, a proper definition for *undesired designs* or in another formidable word the *architectural singularity* seems necessary. Therefore a linear pentapod is an *architecturally singular manipulator* if for every position and orientation, the matrix of Eq. 2.4 becomes rank deficient. By defining the *architecture matrix* of linear pentapods, namely:

$$\mathbf{A} = \begin{pmatrix} r_2 & x_2 & y_2 & z_2 & r_2x_2 & r_2y_2 & r_2z_2 \\ r_3 & x_3 & y_3 & z_3 & r_3x_3 & r_3y_3 & r_3z_3 \\ r_4 & x_4 & y_4 & z_4 & r_4x_4 & r_4y_4 & r_4z_4 \\ r_5 & x_5 & y_5 & z_5 & r_5x_5 & r_5y_5 & r_5z_5 \end{pmatrix}, \tag{2.6}$$

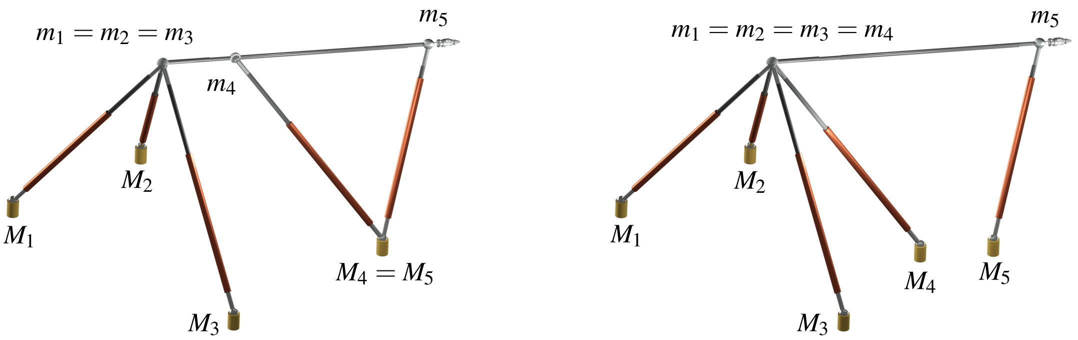


FIGURE 2.2: Illustrations of the counter examples for the necessity of the condition given in Lemma 4.

we can identify such singularities by considering the rank deficiency of this matrix (obtained from the last four rows of the matrix **S**, Eq. 2.4).

Lemma 4. *If the "architecture matrix" Eq. 2.6 is rank deficient then the linear pentapod is an "architecturally singular manipulator" (cf. [Borràs, Thomas, and Torras, 2011a]).*

Remark 6. *It is noteworthy that this is a sufficient but not necessary condition as it is well-known (cf. items (c) and (d) of Corollary 1 in [Nawratil, 2009]) that there exist the following two*[1] *exceptional cases:*

- ⋆ $m_1 = m_2 = m_3$ *and* $M_4 = M_5$,
- ⋆ $m_1 = m_2 = m_3 = m_4$,

which are illustrated in Fig. 2.2. It should be mentioned that these two designs[2] *are equivalent with respect to singularity-invariant leg replacements (cf. [Borràs, Thomas, and Torras, 2011b]).* ◇

Since in computational kinematics most of the computations are of symbolic type, and naturally expensive in the sense of time consumption, it will be highly favourable if one is able to eliminate some extra symbols. The following lemma shows that it is possible to alleviate the burden of extra symbols in computations to come:

Lemma 5. *If the linear pentapod is not architecturally singular then there exists a triple of base points* M_i, M_j *and* M_k *which form a triangle and* $m_i \neq m_j$ *holds (cf. Fig. 2.3).*

Proof. It is enough to show that there is a triangle where at least two of its corresponding platform points are not coinciding. First we claim that a triangle in the base always exist, as otherwise all five base points are collinear which yield a trivial architecture singular design. Now, since not all the platform points can be coincided (if more than 3 platform anchor points coincide we get again a trivial architecture singular design) there should be at least two different points on the platform namely, m_i and m_j. Now name the corresponding base points M_i and M_j. If these two are not coincided then based on the first part of the proof it is possible to find another base point M_k not co-linear with M_i, M_j and hence the statement is fulfilled.

Now, suppose such a triangle with $m_i \neq m_j$ doesn't exist. Then for $\triangle\, M_iM_kM_l$ and $\triangle\, M_iM_kM_l$ we have $m_i = m_k = m_l$ and $m_j = m_k = m_l$ respectively, which would yield $m_i = m_j$, a contradiction. □

[1] Up to renumbering of the platform and base anchor points.

[2] Note that throughout the paper a combinatorial classification [Faugère and Lazard, 1995] of pentapods is used.

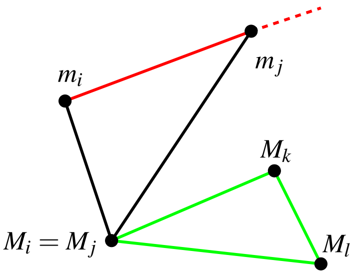


FIGURE 2.3: Red line stands as the linear pentapod's motion platform, while green and black stand for the triangle in the base and the legs respectively.

Based on Lemma 5 one can assume $M_1 = (0,0,0)^T$, $M_2 = (x_2,0,0)^T$ and $M_3 = (x_3,y_3,0)^T$ where $x_2y_3 \neq 0$. Moreover due to $m_1 \neq m_2$ we can assume a scaling upon which, $r_2 = 1$ or $x_2 = 1$ holds. Now the *architecture matrix* of Eq. 2.6 simplifies into the following matrix:

$$\mathbf{A} = \begin{pmatrix} r_2 & x_2 & 0 & 0 & x_2 & 0 & 0 \\ r_3 & x_3 & y_3 & 0 & r_3x_3 & r_3y_3 & 0 \\ r_4 & x_4 & y_4 & z_4 & r_4x_4 & r_4y_4 & r_4z_4 \\ r_5 & x_5 & y_5 & z_5 & r_5x_5 & r_5y_5 & r_5z_5 \end{pmatrix}. \tag{2.7}$$

With the aid of Lemma 5 and using projective geometry it is possible to obtain a simple but helpful geometric interpretation for *architecturally singular* linear pentapods in the coming sections. In fact one can think of $\mathbf{r}$, $\mathbf{X}$ and $\mathbf{r} \circ \mathbf{X}$ as points in the affine space $\mathbb{R}^3$ and the remaining columns of Eq. 2.7 as points on the plane at infinity Ω_∞, which closes $\mathbb{R}^3$ projectively; i.e. the columns of Eq. 2.7 can be seen as homogenous point coordinates of the 3-dimensional projective space $\mathbb{PR}^3$. This point of view implies the following lemma:

Lemma 6. *The "architecture matrix" is rank deficient iff the points* r, X, Y, Z, rX, rY *and* rZ *are coplanar in* $\mathbb{PR}^3$.

Naturally, in the case where some columns of the architecture matrix are zero, they will not appear as points in the projective space interpretation.

2.2 Singularity Variety of Linear Pentapods

Kinematical singularities occur whenever the Jacobian matrix $\mathbf{J}$ becomes rank deficient, where $\mathbf{J}$ can be written as follows (cf. [Borràs, Thomas, and Torras, 2010b]):

$$\mathbf{J} = \begin{pmatrix} \mathbf{l}_1 & \dots & \mathbf{l}_5 \\ \hat{\mathbf{l}}_1 & \dots & \hat{\mathbf{l}}_5 \end{pmatrix}^T \text{ with } \mathbf{l}_j = \begin{pmatrix} u_4 + r_ju_1 - x_j \\ u_5 + r_ju_2 - y_j \\ u_6 + r_ju_3 - z_j \end{pmatrix}, \hat{\mathbf{l}}_j = \begin{pmatrix} z_j(u_5 + r_ju_2) - y_j(u_6 + r_ju_3) \\ x_j(u_6 + r_ju_3) - z_j(u_4 + r_ju_1) \\ y_j(u_4 + r_ju_1) - x_j(u_5 + r_ju_2) \end{pmatrix}.$$

This 5×6 Jacobian matrix $\mathbf{J}$ has a rank less than five whenever the determinants of all its 5×5 sub-matrices vanish. So by naming the determinant of the 5×5 sub-matrix, which results from excluding the jth column, with F_j the singularity loci equals $V(F_1, \dots, F_6)$; i.e. the variety of the ideal spanned by the polynomials $F_1, \dots, F_6$. It can easily be checked by direct computations that this variety equals the zero-set of the greatest common divisor F of

$F_1,\dots,F_6$. This singularity polynomial F has the following structure:

$$\begin{aligned} F := &(A_1u_5+A_2u_6){u_1}^2+[(A_3u_4+A_4u_5+A_5u_6+A_6)u_2+(A_7u_4+A_8u_5\\ &+A_9u_6+A_{10})u_3+(A_{11}u_5+A_{12}u_6)u_4+A_{13}{u_5}^2+(A_{14}u_6+A_{15})u_5\\ &+A_{16}{u_6}^2+A_{17}u_6]u_1+(A_{18}u_4+A_{19}u_6+A_{20}){u_2}^2+[(A_{21}u_4+A_{22}u_5\\ &+A_{23}u_6+A_{24})u_3+A_{25}{u_4}^2+(A_{26}u_5+A_{27}u_6+A_{28})u_4+(A_{29}u_6\\ &+A_{30})u_5+A_{31}{u_6}^2+A_{32}u_6]u_2+(A_{33}u_4+A_{34}u_5+A_{35}){u_3}^2+[A_{36}{u_4}^2\\ &+(A_{37}u_5+A_{38}u_6+A_{39})u_4+A_{40}{u_5}^2+(A_{41}u_6+A_{42})u_5+A_{43}u_6]u_3 \end{aligned} \tag{2.8}$$

where the coefficients A_i belong to the ring

$$R := \mathbb{R}\left[x_2,x_3,x_4,x_5,y_3,y_4,y_5,z_4,z_5,r_2,r_3,r_4,r_5\right], \tag{2.9}$$

which evidently makes F a polynomial with the total-degree of 3 belonging to $R\left[u_1,\dots,u_6\right]$. Note that for a specified orientation (u_1,u_2,u_3) the equation $F=0$ determines only a quadric surface $\Omega(u_1,u_2,u_3)$ in the space of positions. This property is of great importance later on.

Remark 7. *It can easily be checked that the polynomial F is identical with the determinant of the 7×7 matrix $\mathbf{S}$ of Eq. 2.4, (cf. [Borràs, Thomas, and Torras, 2010b, Eq. 4]).*

2.2.1 Rational Parametrization of Singularity Locus

For the computation of the rational parametrization of the linear pentapod's singularity variety, we exploit the idea used in [Coste and Moussa, 2015]: By homogenizing the singularity polynomial F of Eq. 2.8 by the extra variable u_0 with respect to the position variables u_4, u_5 and u_6, we obtain a homogeneous polynomial $F_h \in \mathrm{R}(u_1,u_2,u_3)[u_4,u_5,u_6,u_0]$ in the projective 3-space $\mathbb{P}^3$ with homogeneous coordinates $(u_4:u_5:u_6:u_0)$. It turns out that the point B with homogeneous coordinates $(u_1:u_2:u_3:0)$ is a point of the singularity variety; i.e. $\mathsf{B}\in\mathbf{V}(F_h)\subset\mathbb{P}^3$. Note that B is the ideal point of the linear platform ℓ with orientation vector $\mathbf{i}$.

The side condition on the vector $\mathbf{i}=(u_1,u_2,u_3)^T$ to be of unit-length, can be avoided by using the *stereographic parametrization* of the unit-sphere S^2:

$$\mathbf{x}:(t_3,t_4)\mapsto\left(\frac{2\,t_3}{{t_3}^2+{t_4}^2+1},\frac{2\,t_4}{{t_3}^2+{t_4}^2+1},\frac{{t_3}^2+{t_4}^2-1}{{t_3}^2+{t_4}^2+1}\right). \tag{2.10}$$

Based on this we can parametrize the lines of the bundle $\mathscr{B}$ with vertex B in the finite space $\mathbb{R}^3$ of positions with coordinates (u_4,u_5,u_6) as follows:

$$\mathscr{B}:\begin{pmatrix}u_4\\u_5\\u_6\end{pmatrix}=a\,\mathbf{x}(t_3,t_4)+t_1\,\frac{\partial\mathbf{x}(t_3,t_4)}{\partial t_3}+t_2\,\frac{\partial\mathbf{x}(t_3,t_4)}{\partial t_4}. \tag{2.11}$$

Note that the 2-tuple (t_1,t_2) fixes the line of the bundle $\mathscr{B}$ and the parameter a determines the point on this line. By varying $(t_1,t_2)\in\mathbb{R}^2$ and setting $a=0$ one obtains the plane through the origin, which is orthogonal to $\mathbf{i}$.

Plugging $\mathscr{B}(a,t_1,t_2,t_3,t_4)$ into $F=0$ shows that the resulting expression is only linear in a, as the ideal point B is always one of the two intersection points of a line belonging to $\mathscr{B}$ with the quadric $\Omega(\mathbf{x}(t_3,t_4))$. By solving this linear condition we get $a(t_1,t_2,t_3,t_4)$. Now the singular configurations $\mathfrak{X}=(\xi_1,\dots,\xi_6)\in\mathbb{R}^6$ of the linear pentapod can be rationally parametrized by $(\xi_1,\xi_2,\xi_3) := \mathbf{x}(t_3,t_4)$ and

$$\begin{aligned}
\xi_4 &= 2\,\frac{a\left(t_1,t_2,t_3,t_4\right)t_3}{{t_3}^2+{t_4}^2+1} - 2\,\frac{t_1\left({t_3}^2-{t_4}^2-1\right)}{\left({t_3}^2+{t_4}^2+1\right)^2} - 4\,\frac{t_2\,t_3\,t_4}{\left({t_3}^2+{t_4}^2+1\right)^2},\\
\xi_5 &= 2\,\frac{a\left(t_1,t_2,t_3,t_4\right)t_4}{{t_3}^2+{t_4}^2+1} - 4\,\frac{t_1\,t_3\,t_4}{\left({t_3}^2+{t_4}^2+1\right)^2} + 2\,\frac{t_2\left({t_3}^2-{t_4}^2+1\right)}{\left({t_3}^2+{t_4}^2+1\right)^2},\\
\xi_6 &= \frac{a\left(t_1,t_2,t_3,t_4\right)\left({t_3}^2+{t_4}^2-1\right)}{{t_3}^2+{t_4}^2+1} + 4\,\frac{t_1t_3}{\left({t_3}^2+{t_4}^2+1\right)^2} + 4\,\frac{t_2t_4}{\left({t_3}^2+{t_4}^2+1\right)^2}.
\end{aligned} \tag{2.12}$$

This parametrization covers the singular variety with exception of two low-dimensional sub-variety: A missing 3-dimensional sub-variety is defined by the denominator of $a(t_1,t_2,t_3,t_4)$. In this case the residual intersection point $\in \mathbb{R}^3$ of the line belonging to $\mathscr{B}$ with $\Omega(\mathbf{x}(t_3,t_4))$ is not determined uniquely; i.e. the complete line belongs to $\Omega(\mathbf{x}(t_3,t_4))$. As the orientation $(0,0,1)$ cannot be obtained by the *stereographic parametrization*, also the 2-dimensional sub-variety $\Omega(0,0,1)$ is missing.

Moreover for a given singular pose $\mathfrak{X} = (\xi_1,\ldots,\xi_6) \in \mathbb{R}^6$ one can trivially compute $t_1,\ldots,t_4$ in a rational way from $\xi_1,\ldots,\xi_6$, thus the singularity variety is a rational one (according to the Definitions 19, 20 and 21).

2.3 Designs of the Linear Pentapods Implying a Simple Singularity Variety

The goal of this section is to extract information on the possible designs of the linear pentapods which would lead to possessing a simpler singularity variety. Naturally, having a simple singularity variety has numerous advantageous including a simple path planning and path optimization. These merits will be investigated in later chapters.

The computation of each case is based on the elimination of determinants of *undesired* sub-matrices of Eq. 2.4. These sub-matrices are named $\mathbf{S}\,^{j_1,\ldots,j_n}_{i_1,\ldots,i_n}$ where $i_1,\ldots,i_n$ indicates the numbers of the rows and $j_1,\ldots,j_n$ the numbers of the columns, which have to be deleted from the matrix given in Eq. 2.4; e.g. $\mathbf{S}^{4,5,6}_{1,2,3}$ stands for the sub-matrix obtained by removing the 1st, 2nd and 3rd row and 4th, 5th and 6th column. In this section, by resorting to Lemma 5, $r_2 = 1$.

2.3.1 Linear in Position Variables

In this section we determine all non-architectural singular designs, where the singularity polynomial $\det(\mathbf{S}) = 0$ is only linear in position variables. In the following we distinguish between linear pentapods with/without coplanar base anchor points (planar/non-planar case).

Planar case

Assume that the manipulator is planar ($z_4 = z_5 = 0$). Since the desired goal here is to have the linear singularity polynomial in position variables, all the terms containing position variables of degree two should vanish. These terms form a polynomial, which we call the *undesired polynomial* through the remainder of the section. In a more general sense, the *undesired polynomial* is a polynomial which after subtracting it from $\det(\mathbf{S})$ yields a polynomial with the desired property (this property can be linearity in position/orientation variables or quadratic in total).

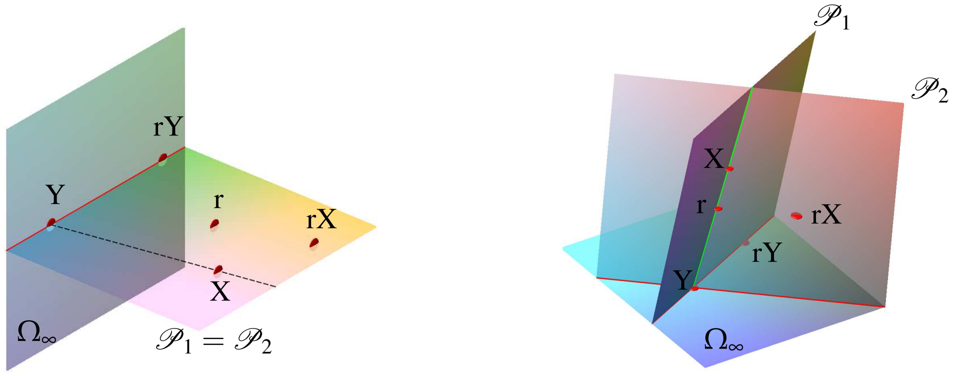


FIGURE 2.4: Geometric interpretation of the conditions yielding a singularity polynomial, which is linear in position variables for a linear pentapod with a planar base: architecturally singular case (left) and the non-architecturally singular case (right).

Here the *undesired polynomial* is as follows:

$$\det\left(\mathbf{S}\,{}^{4,7}_{1,2}\right) u_6^2 + \det\left(\mathbf{S}\,{}^{4,5}_{1,2}\right) u_4 u_6 - \det\left(\mathbf{S}\,{}^{4,6}_{1,2}\right) u_5 u_6 = 0. \tag{2.13}$$

If Eq. 2.13 is fulfilled independently of the position variables then all the coefficients have to be zero. Based on the resulting conditions one can prove the following theorem:

Theorem 3. *A non-architecturally singular linear pentapod with a planar base has a "singularity polynomial" that is linear in position variables, iff there is a singular affine mapping κ from the base plane to the platform line ℓ with $M_i \mapsto m_i$ for $i = 1, \ldots, 5$.*

Proof. Using *Laplace expansion by minors*, $\det\left(\mathbf{S}\,{}^{4,7}_{1,2}\right)$ is:

$$[\mathrm{r}, \mathrm{X}, \mathrm{Y}, \mathrm{rX}]\, u_2 - [\mathrm{r}, \mathrm{X}, \mathrm{Y}, \mathrm{rY}]\, u_1 = 0. \tag{2.14}$$

For all possible orientations, Eq. 2.14 holds whenever both bracket coefficients vanish. Again by considering the *Laplace expansion by minors* for $\det\left(\mathbf{S}\,{}^{4,5}_{1,2}\right)$ and $\det\left(\mathbf{S}\,{}^{4,6}_{1,2}\right)$ respectively, one obtains:

$$[\mathrm{r}, \mathrm{X}, \mathrm{Y}, \mathrm{rY}]\, u_3 = [\mathrm{r}, \mathrm{X}, \mathrm{Y}, \mathrm{rX}]\, u_1 = 0. \tag{2.15}$$

As it is also desired to have these equations vanished for all possible orientations, the bracket coefficients should be equal to zero simultaneously. Hence, independently of all possible orientations, the following statement holds:

$$\det\left(\mathbf{S}\,{}^{4,7}_{1,2}\right) \text{ vanishes} \iff \det\left(\mathbf{S}\,{}^{4,5}_{1,2}\right) \text{ and } \det\left(\mathbf{S}\,{}^{4,6}_{1,2}\right) \text{ vanish.} \tag{2.16}$$

Finally, based on Eq. 2.16 the necessary and sufficient condition for having a singularity polynomial linear in position variables will be:

$$[\mathrm{r}, \mathrm{X}, \mathrm{Y}, \mathrm{rY}] = [\mathrm{r}, \mathrm{X}, \mathrm{Y}, \mathrm{rX}] = 0. \tag{2.17}$$

Using the literature of bracket algebra available at [Ben-Horin et al., 2008; White, 1994] these brackets vanish whenever the four points characterizing them are coplanar. We denote the planes associated with the two brackets of Eq. 2.17 - left and Eq. 2.17 - right by $\mathscr{P}_1$ and $\mathscr{P}_2$, respectively. Then the following two cases have to be distinguished:

1. If the points r, X and Y are not collinear (or in other words if the vectors **r**, **X** and **Y** are linearly independent) then the linear pentapod would be an *architecturally singular*

manipulator since geometrically, by Lemma 6 this is equivalent to having the planes $\mathscr{P}_1$ and $\mathscr{P}_2$ coincident, as depicted in Fig. 2.4 - left.

2. If the points r, X and Y are collinear (or in other words if the vectors $\mathbf{r}$, $\mathbf{X}$ and $\mathbf{Y}$ are linearly dependent) then $\mathbf{r} \in \mathrm{span}\{\mathbf{X},\mathbf{Y}\}$; i.e.

$$\mathbf{r} = \alpha.\mathbf{X} + \beta.\mathbf{Y}, \tag{2.18}$$

where α and β are real numbers. This yields the affine coupling κ mentioned in Theorem 3, namely:

$$\kappa: \ (x_i, y_i) \longmapsto r_i = \alpha x_i + \beta y_i \quad \text{with} \quad \alpha = \tfrac{1}{x_2} \quad \text{and} \quad i = 2, \ldots, 5. \tag{2.19}$$

Geometrically, it is also worth mentioning that the planes $\mathscr{P}_1$ and $\mathscr{P}_2$ do not necessarily coincide in this case (as depicted in Fig. 2.4 - right).

□

Remark 8. *Now, the question arises, which designs possessing a singular affine coupling $\kappa : M_i \mapsto m_i$ are architecturally singular. According to the list given in [Nawratil and Schicho, 2017, Corollary 1] there are three possible types of "architecturally singular manipulators* [3]*", which will be as follows:*

- ⋆ *$m_1, \ldots, m_5$ are distinct and there exists a conic section passing through $M_1, \ldots, M_5$ and the ideal point of the parallel fibers of κ (cf. [Borràs, Thomas, and Torras, 2011a]). This is for example trivially fulfilled if $M_1, \ldots, M_4$ are collinear.*
- ⋆ *$m_1, \ldots, m_4$ are pairwise distinct, $m_4 = m_5$ holds and the base points M_1, M_2, M_3 are collinear.*
- ⋆ *More than two platform points coincide.*

Non-planar case

For the non-planar case the following theorem holds:

Theorem 4. *Non-architecturally singular linear pentapods with a non-planar base possessing a "singularity polynomial", which is linear in position variables, do not exist.*

Proof. $\det(\mathbf{S}) = 0$, independent of the position variables, gives $\det\left(\mathbf{S}\,^{5,2}_{1,2}\right) = \det\left(\mathbf{S}\,^{6,3}_{1,2}\right) = 0$. Using *Laplace expansion by minors* for $\det\left(\mathbf{S}\,^{5,2}_{1,2}\right)$ and $\det\left(\mathbf{S}\,^{6,3}_{1,2}\right)$ one finds:

$$[\mathrm{r}, \mathrm{Y}, \mathrm{Z}, \mathrm{rZ}] = [\mathrm{r}, \mathrm{Y}, \mathrm{Z}, \mathrm{rY}] = 0, \tag{2.20}$$

$$[\mathrm{r}, \mathrm{X}, \mathrm{Z}, \mathrm{rZ}] = [\mathrm{r}, \mathrm{X}, \mathrm{Z}, \mathrm{rX}] = 0. \tag{2.21}$$

Now, it is possible to deduce the following:

1. From Eq. 2.20 - left:

$$\mathbf{r} \circ \mathbf{Z} \in \mathrm{span}\{\mathbf{r}, \mathbf{Y}, \mathbf{Z}\}, \tag{2.22}$$

as $\mathbf{r}$, $\mathbf{Y}$ and $\mathbf{Z}$ are obviously linearly independent.

[3] Up to renumbering of the platform and base anchor points.

2. From Eq. 2.20 - right one derives:

$$\mathbf{r} \circ \mathbf{Y} \in \operatorname{span}\{\mathbf{r}, \mathbf{Y}, \mathbf{Z}\}. \tag{2.23}$$

3. If we replace in Eq. 2.21 - left the expression $\mathbf{r} \circ \mathbf{Z}$ by the linear combination resulting from Eq. 2.22 we get:

$$\mathbf{X} \in \operatorname{span}\{\mathbf{r}, \mathbf{Y}, \mathbf{Z}\}. \tag{2.24}$$

4. From Eq. 2.21 - right and Eq. 2.24 one finds:

$$\mathbf{r} \circ \mathbf{X} \in \operatorname{span}\{\mathbf{r}, \mathbf{Y}, \mathbf{Z}\}. \tag{2.25}$$

Now, using Eqs. 2.22–2.25, four out of seven columns of the *architecture matrix* are linearly dependent and thus rank deficient:

$$\operatorname{Rank}(\mathbf{r}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{r} \circ \mathbf{X}, \mathbf{r} \circ \mathbf{Y}, \mathbf{r} \circ \mathbf{Z}) < 4. \tag{2.26}$$

□

In total the results of Section 2.3.1 show that the manipulator given in Theorem 3 is the only one with a singularity variety linear in position variables. This manipulator design was already known to the authors of [Borràs, Thomas, and Torras, 2011b][4], who also pointed out that the forward kinematics of these pentapods can be solved quadratically.

2.3.2 Linear in Orientation Variables

In this section we determine all non-architecturally singular designs where the singularity polynomial $\det(\mathbf{S}) = 0$ is only linear in orientation variables. As in Section 2.3.1 we distinguish between linear pentapods with planar and non-planar bases.

Planar case

Under the planar condition ($z_4 = z_5 = 0$) the *undesired polynomial* is:

$$\left[\det\left(\mathbf{S}\,^{3,7}_{1,3}\right) + \det\left(\mathbf{S}\,^{4,6}_{1,3}\right)\right] u_2 u_3 + \left[\det\left(\mathbf{S}\,^{2,7}_{1,3}\right) - \det\left(\mathbf{S}\,^{4,5}_{1,3}\right)\right] u_1 u_3 + \\ \det\left(\mathbf{S}\,^{2,6}_{1,3}\right) u_1 u_2 - \det\left(\mathbf{S}\,^{4,7}_{1,3}\right) {u_3}^2 = 0. \tag{2.27}$$

Theorem 5. *A non-architecturally singular linear pentapod with a planar base can only have a "singularity polynomial", which is linear in orientation variables, in the following cases:*

1. *M_2, M_3, M_4, M_5 are collinear,*
2. *$m_1 = m_i$ and M_j, M_k, M_l are collinear with pairwise distinct $i, j, k, l \in \{2,3,4,5\}$,*
3. *$m_1 = m_i = m_j$ with pairwise distinct $i, j \in \{2,3,4,5\}$.*

[4] Note that Theorems 3 and 4 cannot be concluded from Borràs, Thomas, and Torras, 2011b, as the authors restricted to the planar case with no four anchor points aligned.

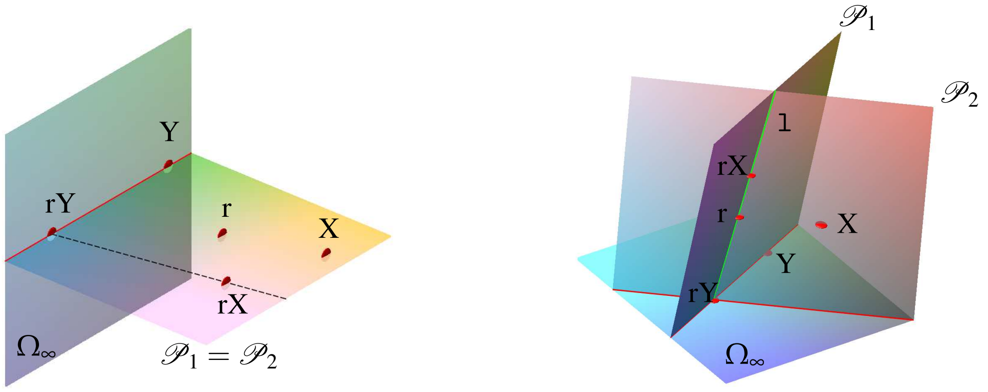


FIGURE 2.5: Geometric interpretation of the conditions yielding a singularity polynomial, which is linear in orientation variables for a linear pentapod with a planar base: architecturally singular case (left) and the non-architecturally singular case (right).

Proof. Eq. 2.27, independently of the orientation variables, gives $\det\left(\mathbf{S}^{\,4,7}_{\,1,3}\right) = 0$. Once again, by resorting to the literature of *brackets*, $\det\left(\mathbf{S}^{\,4,7}_{\,1,3}\right) = 0$ if and only if the following holds:

$$[\mathrm{r},\mathrm{Y},\mathrm{rX},\mathrm{rY}] = [\mathrm{r},\mathrm{X},\mathrm{rX},\mathrm{rY}] = 0. \tag{2.28}$$

Now, name the plane characterized by the points r, rX and rY as $\mathscr{P}_1$. If the points r, rX and rY are not collinear then the plane $\mathscr{P}_1$ is defined *uniquely* and hence by Eq. 2.28 X and Y are also on $\mathscr{P}_1$ which by Lemma 6 results in a rank deficiency of the *architecture matrix* (cf. Fig. 2.5 - left).

On the other hand if the points r, rX and rY are collinear then there is the possibility of having the points X and Y on two different planes, namely $\mathscr{P}_1$ and $\mathscr{P}_2$ as depicted in Fig. 2.5 - right, which does not necessarily lead to an *architectural singularity*. Under this assumption, we get $\mathbf{r} \in \mathrm{span}\{\mathbf{r}\circ\mathbf{X}, \mathbf{r}\circ\mathbf{Y}\}$; i.e.

$$\mathbf{r} = \alpha.\mathbf{rX} + \beta.\mathbf{rY}, \tag{2.29}$$

where α and β are real numbers with $(\alpha,\beta) \neq (0,0)$. Now having Eq. 2.29 in mind, the following possibilities arise (cf. Fig. 2.6):

1. $\forall\, i \in \{2,...,5\}, r_i \neq 0$. This yields:

   $$\mathbf{1} = \alpha.\mathbf{X} + \beta.\mathbf{Y}. \tag{2.30}$$

   Geometrically, in this case, the point $\mathbf{1} = (1,1,1,1)$ should always be on the line $\mathtt{l}$ defined by the two points X and Y. Moreover Eq. 2.30 gives:

   $$\begin{cases} x_2 = \frac{1}{\alpha} \\ \alpha x_i + \beta y_i = 1 \qquad \text{for i} > 2, \end{cases}$$

   which means the base points M_2, M_3, M_4, M_5, are collinear.

2. $\exists!\ i \in \{3,4,5\}$ such that $r_i = 0$. Geometrically this means that one of the points $(1,0,1,1)^T$,$(1,1,0,1)^T$ or $(1,1,1,0)^T$ should be on $\mathtt{l}$. Naturally this yields $m_1 = m_i$ and M_j and M_k are collinear with pairwise distinct $i, j, k \in \{3,4,5\}$.

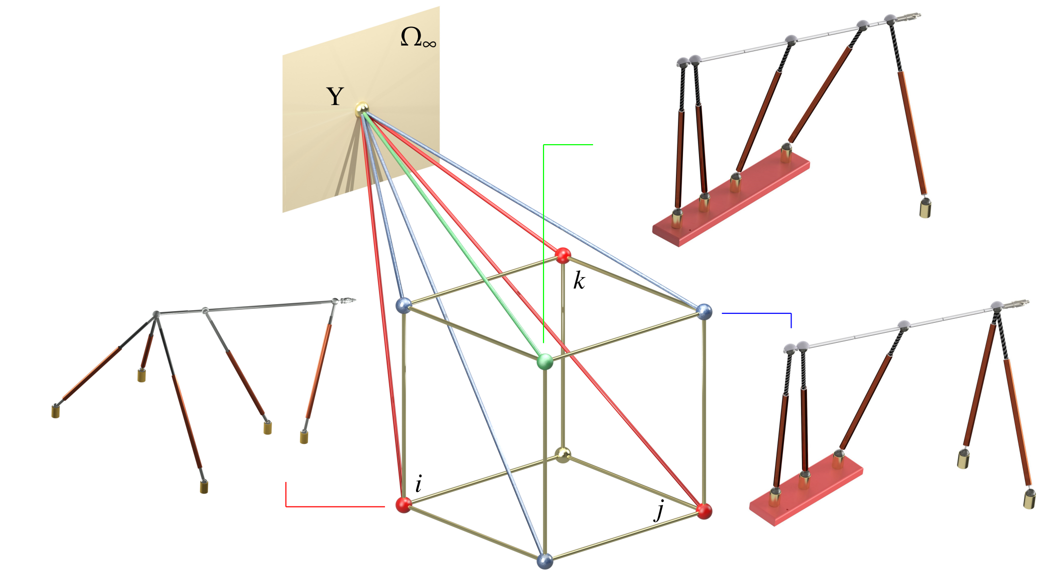


FIGURE 2.6: Three possible designs mentioned in Theorem 5. In this figure, a possible design space is illustrated where the point X can be chosen on the depicted bundle of lines. Blue points stand as the possible designs containing three collinear base points and red points as the design with three collapsed spherical joints while the only green point describes the design with four base points on a line. The collinear base points are indicated by red beams. It is noteworthy that illustrated designs are obtained based on the number of co-incidents (which for red, blue and green are 2,1 and 0 respectively). Additionally, it can be shown by a series of Δ-transforms [Borràs, Thomas, and Torras, 2010b], that the singularity loci of all the three cases are identical.

3. $\exists\, i$ and $j \in \{3,4,5\}$, where $i \neq j$ such that $r_i = r_j = 0$. Geometrically this means that only one of the points $(1,1,0,0)^T$,$(1,0,1,0)^T$ or $(1,0,0,1)^T$ can be on $\mathtt{l}$ which yields $m_1 = m_i = m_j$ with pairwise distinct $i, j \in \{3,4,5\}$.

□

Remark 9. *Note that it is impossible to have the point $(1,0,0,0)^T$ on the line $\mathtt{l}$, since this would mean that the four platform anchor points should be coincident, which yields an "architecturally singular manipulator". For the additional conditions on the designs 1–3 of Theorem 5 rendering the manipulator architecture singular we refer to the list given in [Nawratil and Schicho, 2017, Corollary 1].*

Non-planar case

For the non-planar case the following theorem holds:

Theorem 6. *Non-architecturally singular linear pentapods with a non-planar base possessing a singularity polynomial, which is linear in orientation variables, do not exist.*

Proof. $\det(\mathbf{S}) = 0$, independently of orientation variables, gives $\det\left(\mathbf{S}\,^{2,5}_{1,3}\right) = \det\left(\mathbf{S}\,^{3,6}_{1,3}\right) = \det\left(\mathbf{S}\,^{4,7}_{1,3}\right) = 0$. By using *Laplace expansion by minors* for $\det\left(\mathbf{S}\,^{2,5}_{1,3}\right)$, $\det\left(\mathbf{S}\,^{3,6}_{1,3}\right)$ and $\det\left(\mathbf{S}\,^{4,7}_{1,3}\right)$ one finds:

$$[\mathrm{r}, \mathrm{Y}, \mathrm{rY}, \mathrm{rZ}] = [\mathrm{r}, \mathrm{Y}, \mathrm{rX}, \mathrm{rY}] = 0, \tag{2.31}$$

$$[\mathrm{r}, \mathrm{Z}, \mathrm{rX}, \mathrm{rZ}] = [\mathrm{r}, \mathrm{X}, \mathrm{rX}, \mathrm{rZ}] = 0. \tag{2.32}$$

Now from Eq. 2.31 and Eq. 2.32 the following is argued:

1. From Eq. 2.31 - left:
$$\mathbf{r}\circ\mathbf{Y}\in\operatorname{span}\{\mathbf{r},\mathbf{Y},\mathbf{r}\circ\mathbf{Z}\}. \tag{2.33}$$
2. From Eq. 2.31 - right and Eq. 2.33:
$$\mathbf{r}\circ\mathbf{X}\in\operatorname{span}\{\mathbf{r},\mathbf{Y},\mathbf{r}\circ\mathbf{Z}\}. \tag{2.34}$$
3. From Eq. 2.32 - left and Eq. 2.34:
$$\mathbf{Z}\in\operatorname{span}\{\mathbf{r},\mathbf{Y},\mathbf{r}\circ\mathbf{Z}\}. \tag{2.35}$$
4. From Eq. 2.32 - right and Eq. 2.34:
$$\mathbf{X}\in\operatorname{span}\{\mathbf{r},\mathbf{Y},\mathbf{r}\circ\mathbf{Z}\}. \tag{2.36}$$

Now, considering Eqs. 2.33–2.36 one would obtain
$$\operatorname{Rank}(\mathbf{r},\mathbf{X},\mathbf{Y},\mathbf{Z},\mathbf{r}\circ\mathbf{X},\mathbf{r}\circ\mathbf{Y},\mathbf{r}\circ\mathbf{Z})<4. \tag{2.37}$$
which implies an *architecturally singular manipulator*. □

2.3.3 Quadratic

In this section we study linear pentapods where the singularity polynomial is only quadratic in total. We are only able to prove the following negative result:

Theorem 7. *Non-architecturally singular linear pentapods possessing a singularity polynomial, which is quadratic in pose variables, do not exist.*

Proof. Despite the fact that it is possible to prove this theorem in a similar fashion as the proofs of the Theorem 4 and Theorem 6, here we choose a different constructive-computational approach. The singularity polynomial given in Eq. 2.8 is a polynomial in pose variables. However, the coefficients of this polynomial contain architectural information regarding the geometric structure of the pentapod. Hence, rewriting Eq. 2.8 in the following form

$$\begin{aligned} F =\ &(A_1 u_2 + A_2 u_3)u_4{}^2 + ((A_3 u_1 + A_4 u_2 + A_5 u_3)u_5 + (A_6 u_1 + A_7 u_2 + A_8 u_3)u_6 + \\ &(A_9 u_2 + A_{10} u_3)u_1 + A_{11} u_2{}^2 + (B_1 + A_{12} u_3)u_2 + A_{13} u_3{}^2 + B_2 u_3)u_4 + \\ &(A_{14} u_1 + A_{15} u_3)\, u_5{}^2 + ((A_{16} u_1 + A_{17} u_2 + A_{18} u_3)u_6 + A_{19} u_1{}^2 + (B_3 + \\ &A_{20} u_2 + A_{21} u_3)u_1 + (B_4 + A_{22} u_3)u_2 + A_{23} u_3{}^2 + B_5 u_3)\, u_5 + (A_{24} u_1 + \\ &A_{25} u_2)\, u_6{}^2 + (A_{26} u_1{}^2 + (B_6 + A_{27} u_2 + A_{28} u_3)u_1 + A_{29} u_2{}^2 + (B_7 + \\ &A_{30} u_3)\, u_2 + B_8 u_3)\, u_6 \end{aligned} \tag{2.38}$$

implies that if we are about to carve a quadratic polynomial out, then the variables given in magenta should not vanish all at the same time while the rest of coefficients (given by A_i for $i = 1,\ldots,30$) are the ones that must vanish simultaneously. Note that, in the light of Lemma 5, A_i and B_j should belong to the ring $R := \mathbb{R}[r_3,r_4,r_5,x_2,x_3,x_4,x_5,y_3,y_4,y_5,z_4,z_5]$ (in fact this is the least number of variables we can resort to in a general setting). This yields a rather famous algebraic geometry problem known as the *ideal membership problem*. In fact, if we define the ideal
$$I := \langle A_1, A_2, \ldots, A_{30}\rangle \lhd R, \tag{2.39}$$
the variety $V := \mathbf{V}(I)$ is a 9-dimensional algebraic variety in $\mathbb{R}^{12}$ (which we can think of it as a sort of design space where each point of it represents a possible architectural design for

the pentapod). Thinking of the polynomials B_j (for $j = 1,\ldots,8$) as maps between $\mathbb{R}^{12}$ and $\mathbb{R}$, we would like to know whether they all simultaneously vanish over V or not. By *Hilbert's Nullstellensatz theorem* (see Theorem 1) we know that the set of all polynomial functions that vanish on V is $\sqrt{I}$. Hence the whole problem reduces to knowing whether all B_js belong to $\sqrt{I}$ simultaneously or not. Using the following command in Maple

```
for j from 1 to 8 do RadicalMembership(B[j],I) end do;
```

reveals that the statement is true for all 8 cases. Consequently, all A_is vanishing implies that all B_js vanish which results in an architecturally singular manipulator. □

2.4 Simple Pentapods

The results of the previous section paves the way for the introduction of a new family of linear pentapods possessing a singularity variety linear in position/orientation variables. Due to the findings in Theorem 3 and Theorem 5, the aforementioned family has to be a subclass of linear pentapods with a planar base.

Definition 27. *A planar pentapod with an architectural design resulting in a singularity polynomial linear in orientation/position variables (LO/LP) is called a "simple pentapod".*

It turns out that the singularity variety of *simple pentapods* has some certain interesting topological and geometrical features to study. In the following subsection these properties are exploited.

2.4.1 Σ - Variety

Definition 28. *The locus of the singularity polynomial of a "simple pentapod" is called a "Σ-variety".*

Now, tidied up the known information through definitions, one can take a closer look at the structure of the Σ-varieties of LO and LP-cases.

Lemma 7. *The Σ-variety of the LO-cases is the zero set of the following polynomial:*

$$\Sigma : u_6\ \left[u_6(\alpha u_1+\beta u_2)-u_3(\alpha u_4+\beta u_5-1)\right]=0, \tag{2.40}$$

where α and β are real numbers and $\alpha^2+\beta^2\neq 0$ (cf. [Rasoulzadeh and Nawratil, 2019a]).

Proof. It can be shown by a series of Δ-transforms (cf. [Borràs, Thomas, and Torras, 2010b]), that the singularity loci of all three cases are identical. Now, by substituting the relations between architecture parameters of one of the cases (e.g. 3rd-case identified by pairwise distinct indices $i,j,k\in\{3,4,5\}$) into $\det(\mathbf{S})$ (cf. Eq. 2.4):

$$x_2 := \frac{1}{\alpha},\qquad x_i := \frac{1-\beta y_i}{\alpha},\qquad r_j = r_k := 0, \tag{2.41}$$

one obtains Eq. 2.40. □

Lemma 8. *The Σ-variety of the LP-case is the zero set of the following polynomial:*

$$\Sigma : u_3\ \left[u_6(\alpha u_1+\beta u_2-1)-u_3(\alpha u_4+\beta u_5)\right]=0. \tag{2.42}$$

Proof. From Theorem 3, it is known that LP-case is generated whenever there is a *singular affine map* $\kappa : (x_i,y_i)\longrightarrow r_i$, where $r_i=\alpha x_i+\beta y_i$ (cf. Eq. 2.19). Substituting this relation into $\det(\mathbf{S})$ and factorizing, one obtains Eq. 2.42. □

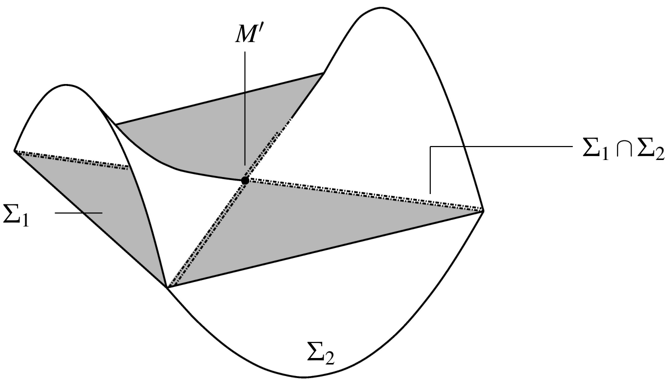


FIGURE 2.7: One way to have an imaginative illustration of the Σ-variety is to consider a case with a certain similarity in $\mathbb{R}^3$. In order to do so, consider the variety $\mathbf{V}\left(z\left(x^2-y^2-z^2\right)\right)$ which is in fact the union of a plane Σ_1 and a hyperbolic paraboloid Σ_2. In that way, the $\Sigma_1 \cap \Sigma_2$ will be of co-dimension 2 while Σ_1 and Σ_2 are tangent at a 0 dimensional subvariety. Note that due to the fact that in this example we are restricted to $\mathbb{R}^3$ it is impossible to show the 2-dimensional singular set on Σ_2 as it is of co-dimension 4.

Later on, during the process of singularity-free path optimization, a projection of a point in $\mathbb{R}^6$ onto the variety is needed. Hence, a more detailed understanding of the Σ-variety is helpful. In the following theorem the properties of the Σ-variety in the *real space* are investigated.

Theorem 8. *The Σ-variety has the following properties:*

a) It is an algebraic variety formed by the union of a hyperplane Σ_1, and a hyperquadric Σ_2, in $\mathbb{R}^6$,

b) $\Sigma_2 = \Sigma_3 \uplus M$, where Σ_3 is the set of singular points of Σ_2 and M is a smooth manifold,

c) $\Sigma_1 \cap \Sigma_2$ is a 4-dimensional algebraic variety, consisting of the union of two 4-planes $\mathscr{A}$ and $\mathscr{B}$,

d) Σ_1 is tangent to Σ_2 at a 3-dimensional smooth manifold $M' \subset \Sigma_1 \cap \Sigma_2$,

e) Σ_3 is a 2-plane and is contained in $\mathscr{A} \cap \mathscr{B} \subset \Sigma_1 \cap \Sigma_2$.

Finally, having in mind that $\mathbf{V}(f_1,\dots,f_n)$ is the set of solutions to the system of polynomial equations $\langle f_1,\dots,f_n\rangle$, the details are summarized below:

- *LO-case :*

$$\Sigma_1 = \mathbf{V}(u_6),$$
$$\Sigma_2 = \mathbf{V}\left(u_6(\alpha u_1 + \beta u_2) - u_3(\alpha u_4 + \beta u_5 - 1)\right),$$
$$\Sigma_3 = \mathbf{V}(\alpha u_1 + \beta u_2, u_3, \alpha u_4 + \beta u_5 - 1, u_6).$$

- *LP-case:*

$$\Sigma_1 = \mathbf{V}(u_3),$$
$$\Sigma_2 = \mathbf{V}\left(u_6(\alpha u_1 + \beta u_2 - 1) - u_3(\alpha u_4 + \beta u_5)\right),$$
$$\Sigma_3 = \mathbf{V}\left(\alpha u_1 + \beta u_2 - 1, u_3, \alpha u_4 + \beta u_5, u_6\right).$$

Proof. Since proving the theorem for one of the LP/LO cases can easily be repeated for the other case, without loss of generality, we prove the theorem for the “LO-case":

a) Using Lemma 7 and basic properties of algebraic varieties (cf. [Cox, Little, and O'Shea, 2013]) results in:

$$\Sigma = \mathbf{V}(u_6) \cup \mathbf{V}(\, u_6\, (\alpha\, u_1 \,+\, \beta\, u_2) \,-\, u_3\, (\, \alpha\, u_4 \,+\, \beta u_5 \,-\, 1\,)\,) = \Sigma_1 \cup \Sigma_2. \tag{2.43}$$

b) The set of all singular points of Σ_2 are the solutions of $\langle \frac{\partial f}{\partial u_1}, \frac{\partial f}{\partial u_2}, \frac{\partial f}{\partial u_3}, \frac{\partial f}{\partial u_4}, \frac{\partial f}{\partial u_5}, \frac{\partial f}{\partial u_6}, f \rangle$ where $f \in \mathbb{C}[u_1, u_2, u_3, u_4, u_5, u_6]$. If f is Eq. 2.40 then

$$\Sigma_3 = \mathbf{V}\,(\, u_6,\, 1-(\alpha u_4+\beta u_5),\, u_3,\, \alpha\, u_1+\beta\, u_2\,)\,. \tag{2.44}$$

Considering the following evidently smooth maps

$$h_1 : \mathbb{R}^5 \setminus \mathbf{V}\,(\,\alpha\, u_1+\beta\, u_2\,) \longrightarrow \mathbb{R}^6, \tag{2.45}$$

$$(u_1,u_2,u_3,u_4,u_5) \longmapsto \left(u_1,u_2,u_3,u_4,u_5, \frac{u_3(\alpha u_4+\beta u_5-1)}{\alpha\, u_1+\beta\, u_2} \right), \tag{2.46}$$

$$h_2 : \mathbb{R}^5 \setminus \mathbf{V}\,(\,1-(\alpha u_4+\beta u_5)\,) \longrightarrow \mathbb{R}^6, \tag{2.47}$$

$$(u_1,u_2,u_4,u_5,u_6) \longmapsto \left(u_1,u_2, \frac{u_6\,(\alpha\, u_1+\beta\, u_2)}{\alpha\, u_4+\beta\, u_5-1}, u_4,u_5,u_6 \right), \tag{2.48}$$

$$h_3 : \mathbb{R}^5 \setminus \mathbf{V}\,(\,u_3\,) \longrightarrow \mathbb{R}^6, \tag{2.49}$$

$$(u_1,u_2,u_3,u_5,u_6) \longmapsto \left(u_1,u_2,u_3, \frac{\alpha\, u_1u_6+\beta\, u_2u_6-\beta\, u_3u_5+u_3}{u_3\alpha}, u_5,u_6 \right), \tag{2.50}$$

$$h_4 : \mathbb{R}^5 \setminus \mathbf{V}\,(\,u_6\,) \longrightarrow \mathbb{R}^6, \tag{2.51}$$

$$(u_2,u_3,u_4,u_5,u_6) \longmapsto \left(\frac{\alpha\, u_3u_4-\beta\, u_2u_6+\beta\, u_3u_5-u_3}{u_6\alpha}, u_2,u_3,u_4,u_5,u_6 \right), \tag{2.52}$$

gives 5-dimensional smooth manifolds $M_i :=$ graph$\,(h_i) \subset \mathbb{R}^6$ whose intersection with Σ_3 is empty. Moreover, taking an element in Σ_2 and noting the structure of h_i domains implies one of the following two possibilities:

- if it satisfies Eq. 2.44 then it belongs to Σ_3,
- otherwise it belongs to at least one of the M_is.

Hence, by naming $M := \bigcup_{i=1}^{4} M_i$ we get the required result.

c) Substituting $u_6 = 0$ in the defining equation of the hypersurface gives

$$\Sigma_1 \cap \Sigma_2 = \mathbf{V}\,(u_3) \cup \mathbf{V}(\alpha u_4+\beta u_5-1) = \mathscr{A} \cup \mathscr{B}, \tag{2.53}$$

which upon observing that hypersurface equation and Eq. 2.53 are two constraints for pose variables or through direct computation (i.e. using HilbertDimension command in Maple) reveals that $\Sigma_1 \cap \Sigma_2$ is a union of two 4-dimensional planes.

d) Assuming f_1 and f_2 to be the defining polynomials of Σ_1 and Σ_2, to obtain the tangency one has to check the points of $\Sigma_1 \cap \Sigma_2$ in which ∇f_1 and ∇f_2 are collinear. Doing so yields M' with the following parametrization:

$$\mathbb{R}^3 \setminus \mathbf{V}\,(\alpha\, v_1 + \beta\, v_2) \longrightarrow \mathbb{R}^6, \tag{2.54}$$

$$(v_1, v_2, v_3) \longmapsto \left(v_1, v_2, 0, \frac{1-\beta\, v_3}{\alpha}, v_3, 0\right), \tag{2.55}$$

for the LO-case which trivially yields a 3-dimensional smooth manifold. Using the above parametrization (or through direct manipulation of the corresponding implicit equations) one finds that Σ_3 is a 2-dimensional plane and a subset of $\Sigma_1 \cap \Sigma_2$. By observing Eq. 2.53, one deducts $\langle u_3, (\alpha\, u_4 + \beta\, u_5 - 1)\rangle$ includes $\langle u_3\ (\alpha\, u_4 + \beta\, u_5 - 1)\rangle$ which results in $\Sigma_3 \subset \mathscr{A} \cap \mathscr{B}$.

e) The parametrization of Σ_3 for LO-case is:

$$(v_1, v_2) \longmapsto \left(-\frac{\beta\, v_1}{\alpha}, v_1, 0, \frac{1-\beta\, v_2}{\alpha}, v_2, 0\right). \tag{2.56}$$

□

In Fig. 2.7, an imaginative geometric depiction of Σ-variety in $\mathbb{R}^6$ is described in such a way that the there is a correspondence between the co-dimension of its properties and of the properties listed in Theorem 8.

2.5 Kinematic Redundant Designs

A certain drawback of parallel robots is the limitation of their singularity-free workspace, which can be overcome by the concept of redundancy. A review on the different types of redundancy for parallel robots with S$\underline{\text{P}}$S-legs including a discussion of their pros and cons is given by the authors in [Nawratil and Rasoulzadeh, 2018]. Following these arguments, we are preferring the concept of kinematic redundancy by reconfiguring the base anchor points of the pentapod by additional joints. In the case of a given path of the platform, the kinematic redundant DoFs can be used to avoid singularities (if possible[5]) and to increase the performance of the manipulator during the prescribed motion [Borràs et al., 2009].

For kinematic redundant designs of the simple pentapods the optimal reconfiguration of the base (regarding the distance from the singularity) can easily be done due to the above mentioned closed form solution (will be discussed in Theorem 9 and Section 3.3.3). Therefore we conclude the chapter by presenting three such designs. The designs proposed in Sections 2.5.1 and 2.5.2 have two DoFs of kinematically redundancy and the design given in Section 2.5.3 has even three kinematic redundant DoFs.

2.5.1 Design 1

This design, displayed in Fig. 2.8, is based on the idea to change the coefficient β of the affine coupling κ given in Eq. 2.19 by a reconfiguration of the base. This can be achieved by a suitable sliding of the base points. The fibers of the singular affine transformation κ from the base plane to the platform correspond to parallel lines in the base plane. It is well known (cf. Section 4.3 of [Borràs, Thomas, and Torras, 2011a]) that a reconfiguration of a base point along its corresponding fiber does not change the singularity variety. Therefore it suggests itself to mount the sliders orthogonal to the fiber-direction. This sliding gives the first degree of kinematic redundancy.

[5] The singularity variety is a hypersurface in the mechanism's configuration space; thus two points of the configuration space can be separated by this hypersurface.

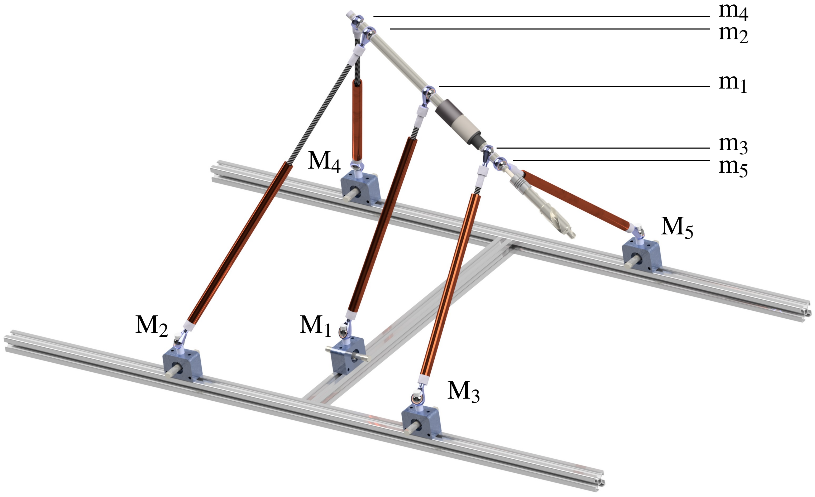


FIGURE 2.8: Illustration of the kinematic redundant linear pentapod of Section 2.5.1 with a linear singularity variety in position variables.

Remark 10. *The linear pentapod given in Fig. 2.8 has been designed in a symmetric way, such that the sliders of M_i and M_{i+1} (for $i = 2,4$) have to move with the same velocity (but in opposite directions). Note that one can drive all sliders of $M_2,\dots,M_5$ with only one motor and a fixed gearing, as the ratio of the velocities of the sliders of M_2 and M_4 is constant.*

Moreover it can easily be checked, that the symmetric design proposed in Fig. 2.8, can never be architecturally singular in practice. ⋄

The second degree of kinematic redundancy is achieved by the sliding of the first base point in fiber-direction. As already mentioned this will not affect the singularity surface, but it can be used to increase the performance of the manipulator during an end-effector motion [Borràs et al., 2009].

2.5.2 Design 2

This design, based on item 1 of Theorem 5 and displayed in Fig. 2.9, is also a 2-DoF kinematic redundant pentapod with a planar base, which has the property that its singular polynomial is linear in orientation for all possible configurations. The base points $M_2,\dots,M_5$ are collinearly mounted on a rod g, which slides (active joint) along a circular rail on the ground and is connected over a U-joint (passive joint) with the ceiling. Therefore the rod g generates during the motion a right circular cone.

For a better understanding of the redundant DoFs, we have a look at the singular-invariant replacement of legs keeping the given platform anchor points:

⋆ As this linear pentapod contains a line-line component (cf. [Borràs, Thomas, and Torras, 2010b]), one can relocate the base anchor points of the legs $m_2M_2,\dots,m_5M_5$ arbitrarily on g (assumed that the resulting manipulator is not architecturally singular).

  Remark 11. *One can additionally allow a sliding (by active joints) of the base points along the rod g (yielding further degrees of kinematic redundancy) but this will not change the singularity variety. These reconfigurations can only be used to improve the performance of the manipulator.* ⋄

⋆ The base point of the first leg can be replaced by any point of the plane spanned by M_1 and g (assumed that the resulting manipulator is not architecturally singular). Therefore a sliding of M_1 along the circular rail changes the singularity variety.

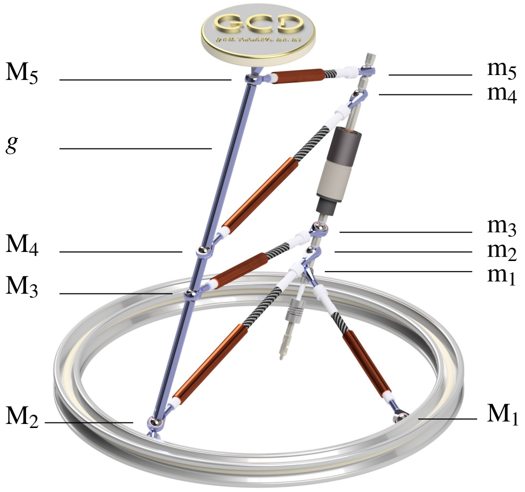


FIGURE 2.9: Illustration of the kinematic redundant linear pentapod of Section 2.5.2 with a linear singularity variety in orientation variables. The suggested design, where the upper part is mounted on the ceiling, can be of interest for e.g. the milling of an object without any need of its repositioning, as the manipulator can go around the object by 360 degrees.

2.5.3 Design 3

This design, based on item 2 of Theorem 5 and displayed in Fig. 2.10, is a 3-DoF kinematically redundant pentapod with planar base, which has the property that its singular polynomial is linear in orientation for all possible configurations. The anchor points M_1 and M_2 can slide along a circular rail (two active joints). The third degree of kinematic redundancy is obtained by the rotation of the rod ℓ on which the collinear points M_3, M_4, M_5 are mounted.

For a better understanding of the redundant DoFs, we study again the singular-invariant replacements of legs keeping the given platform anchor points:

- ⋆ One can relocate the base anchor points of the legs m_3M_3, m_4M_4, m_5M_5 arbitrarily on g (assumed that the resulting manipulator is not architecturally singular). Therefore also Remark 11 holds in this context.
- ⋆ The base points of the first and second leg can be replaced by any two points of the carrier plane of the circular rail (assumed that the resulting manipulator is not architecturally singular). As a consequence the sliding of M_1 and M_2 along the circular rail does not change the singularity variety. Therefore these two redundant DoFs can only be used to improve the performance of the manipulator.

Remark 12. *Finally it should be noted that a design, based on item 3 of Theorem 5, is not suited for technical realization due to the triple joint at the platform.* ◇

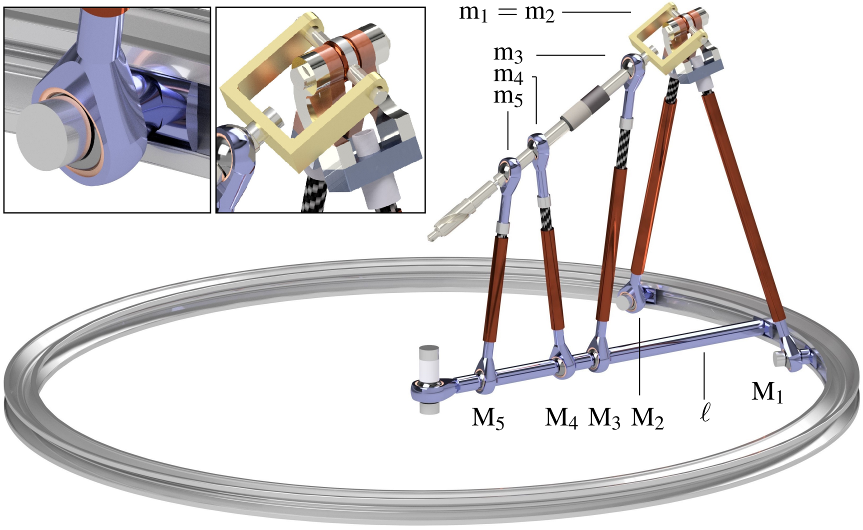


FIGURE 2.10: Illustration of the kinematic redundant linear pentapod of Section 2.5.3 with a linear singularity variety in orientation variables. This design also allows a milling by 360° degrees around the object. Moreover, detailed views of the double joint $m_1 = m_2$ at the platform and the circular slider of M_2 are provided.

Chapter 3

Distance to the Singularity Variety

In singularities the number of DoFs of the mechanism changes instantaneously and becomes uncontrollable. Additionally the actuator forces can become very large and cause the break down of the platform [Li, Gosselin, and Richard, 2007]. Henceforth knowing the distance of a given pose from the singularity variety is of great importance. The chapter is designed as follows. We start by a lemma which reformulated the object-oriented metric tensor into a metric. This function allows us to interpret the distance between two points in the pose space (in $\mathbb{R}^6$) as the distance between two configurations of the pentapod (in $\mathbb{R}^3$). The next pace thoroughly investigates the distance to the singularity variety for the linear pentapods and simple pentapods (cf. Definition 27) for a fixed orientation/position under Euclidean and equiform motions. Finally, the results of these investigations are summarized in a table given at the end of the chapter.

3.1 Object-oriented Metric

In this section the goal is to introduce the object-oriented metric which is of special interest as it allows one to deal with the kinematic space in a more general format (without forcing us to restrict ourselves to position or orientation space). In order to introduce this metric we define an object-oriented metric tensor and from that we derive the metric function.

Definition 29. *The“object-oriented metric tensor" relative to the canonical basis of $\mathbb{R}^6$ is defined by the following matrix:*

$$\mathbf{g} = \begin{pmatrix} R & & & J & & \\ & R & & & J & \\ & & R & & & J \\ J & & & 1 & & \\ & J & & & 1 & \\ & & J & & & 1 \end{pmatrix}_{6\times 6}, \tag{3.1}$$

where $R = \frac{1}{5}(r_1^2 + r_2^2 + r_3^2 + r_4^2 + r_5^2)$ and $J = \frac{1}{5}(r_1 + r_2 + r_3 + r_4 + r_5)$.

Before going further the following lemma is crucial in the coming interpretations:

Lemma 9. *The object-oriented metric can be reformulated into*

$$\mathfrak{d}(p,u)^2 := \frac{1}{5}\sum_{i=1}^{5} \|m_i - m_i^{'}\|^2, \tag{3.2}$$

where p and u are two configurations and m_i and $m_i^{'}$ denote the coordinate vectors of the corresponding platform anchor points [Nawratil, 2017] (cf. Fig. 3.1).

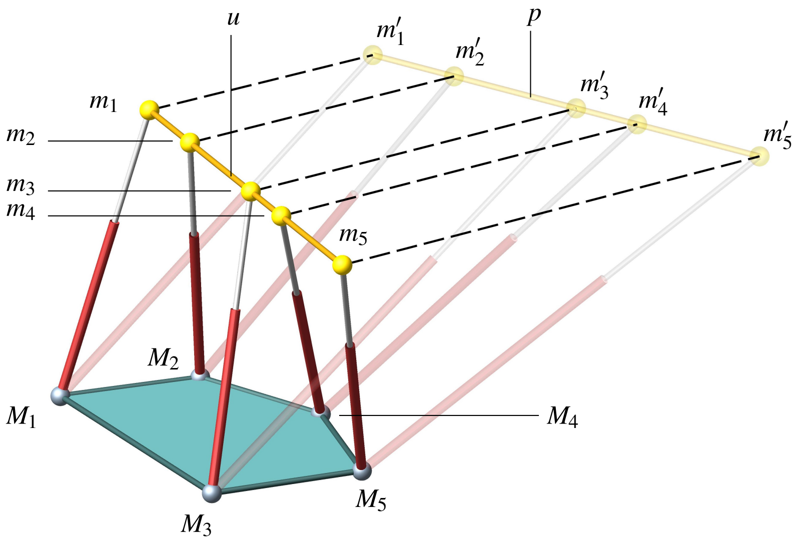


FIGURE 3.1: Illustration of a linear pentapod with a planar base. From a geometrical point of view, the value of Eq. 3.2, for the two poses of the planar pentapod, is equal to $1/5$ of the sum of the squared lengths of the dashed lines between two platforms configurations.

Proof. Define $m_i := (u_4, u_5, u_6)^T + r_i\,(u_1, u_2, u_3)^T$ and $m_i' := (p_4, p_5, p_6)^T + r_i\,(p_1, p_2, p_3)^T$ for $i = 1, \dots, 5$ as the platform anchor points. Then expanding the right hand side of Eq. 1.12 gives Eq. 3.2. □

In kinematics, especially in the area of path planning, it is quite usual to decompose a manipulators motion into the fixed orientation, (namely all the positions the end effector can attain) and the fixed position case (all the orientations a manipulator attains) at each stance of the motion [Li et al., 2017]. However, as it is inferred from Fig. 3.1, Lemma 3.2 allows us to interpret the distance between two points p and $u \in \mathbb{R}^6$ as the distance between two configurations in $\mathbb{R}^3$. Consequently, this interpretation alleviates the burden of such a decomposition and lets us to deal with the entire motion as a curve in either $\mathbb{R}^6$ or $\mathbf{S}^2 \times \mathbb{R}^3$ in one go. However, for the sake of simplicity in following the text, the corresponding metric spaces to be investigated in this chapter are summarized in Table 3.1. In the coming subsections Euclidean and equiform motions are thoroughly investigated for the linear pentapods.

In all the cases to come assume that the given pose is $\mathfrak{G} = (p_1, p_2, \dots, p_6)$. Additionally, the coordinates of the pose space are assumed to be $(u_1, u_2, \dots, u_6)$.

Metric Spaces		
Motion	Space	Metric
Positions	$\mathbb{R}^3$	*ordinary Euclidean metric*
Orientations	$\mathbf{S}^2$	*round metric*
Euclidean motion	$\mathbf{S}^2 \times \mathbb{R}^3 \subset \mathbb{R}^6$	*object-oriented metric (from* $\mathbb{R}^6$*)*
Equiform motion	$\mathbb{R}^6$	*object-oriented metric*

TABLE 3.1: A summary of metric spaces of interest to the possible motions of a linear pentapod. Note that the translations and orientations are referring to fixed orientation and fixed position motions respectively while Euclidean motion stands as rigid body transformations. Equiform motions are the rigid body motions which do not preserve the size of the object.

3.2 General Linear Pentapods

3.2.1 Positions (Fixed Orientation Case)

We ask for the closest singular configuration $\mathfrak{O}$ having the same orientation (p_1, p_2, p_3) as the given pose $\mathfrak{G}$. As $\mathfrak{G}$ and $\mathfrak{O}$ only differ by a translation, one could define the distance between these two poses by the length of the translation vector. This is equal to restricting our case to $\mathbb{R}^3$ with *ordinary Euclidean metric* (cf. Table 3.1). Therefore $\mathfrak{O}$ has to be a pedal point on $\Omega(p_1, p_2, p_3)$ with respect to the point (p_4, p_5, p_6). The set $\mathscr{O}$ of all these pedal points equals the variety $V(\frac{\partial L}{\partial u_4}, \frac{\partial L}{\partial u_5}, \frac{\partial L}{\partial u_6}, \frac{\partial L}{\partial \lambda})$ where λ is the Lagrange multiplier of the Lagrange equation

$$L(u_4, u_5, u_6, \lambda) = (u_4 - p_4)^2 + (u_5 - p_5)^2 + (u_6 - p_6)^2 + \lambda F, \tag{3.3}$$

where F is the singularity polynomial obtained from Eq. 2.8. It turns out that the equations $\frac{\partial L}{\partial u_4} = \frac{\partial L}{\partial u_5} = \frac{\partial L}{\partial u_6} = 0$ are linear with respect to u_4, u_5 and u_6. Solving these equations for these variables and plugging in the obtained expressions into $\frac{\partial L}{\partial \lambda} = 0$ gives a polynomial, which has 354513 terms, and is of degree 6 in λ. Hence, $\mathscr{O}$ consists of six points over $\mathbb{C}$, where the closest one to (p_4, p_5, p_6) implies $\mathfrak{O}$ (see Figs. 3.2 - left and 3.3 - left).

Example 20. *Assume a general linear pentapod with the following architecture matrix:*

$$\begin{pmatrix} r_2 & x_2 & y_2 & z_2 \\ r_3 & x_3 & y_3 & z_3 \\ r_4 & x_4 & y_4 & z_4 \\ r_5 & x_5 & y_5 & z_5 \end{pmatrix} = \begin{pmatrix} 2 & 5 & 0 & 0 \\ 4 & -4 & -3 & 0 \\ 5 & 3 & 7 & -6 \\ 10 & 9 & -5 & 4 \end{pmatrix}. \tag{3.4}$$

It turns out that for a random example (i.e. pose $\mathfrak{G}$ is given in the caption of Fig. 3.2 - left) one obtains 4 real solutions and 2 complex ones. The corresponding values of u_4, u_5, u_6 are obtained by back-substitution (cf. Table 3.2).

	u_4	u_5	u_6	λ	l
1	2.477488952	2.697875818	0.9626913680 e-1	0.7113733065 e-4	3.944412424
2	1.947281086	-12.89155068	4.083319654	0.1588929873 e-2	15.89185655
3	-6.068046868	-11.26894102	1.797066843	0.2894577270 e-2	16.53931597
4	23.47334506	-5.118737128	5.125091422	0.5105991308 e-2	22.98443541

TABLE 3.2: The 4 real solutions in ascending order with respect to the length l of the translation vector towards the given position.

3.2.2 Orientations (Fixed Position Case)

On another hand one can ask for the closest singular configuration $\mathfrak{P}$, which has the same position (p_4, p_5, p_6) as the given pose $\mathfrak{G}$. As $\mathfrak{G}$ and $\mathfrak{P}$ only differ in orientation, the angle $\in [0, \pi]$ enclosed by these two directions can be used as distance function. Note that this angle is the Riemannian distance on $\mathbf{S}^2$ (cf. Section 1.1.5 and Table 3.1).

By intersecting the singularity variety Σ, for the given position (p_4, p_5, p_6), with $\mathbf{S}^2$ one obtains a spherical curve $\omega(p_4, p_5, p_6)$ of degree 4. Then $\mathfrak{P}$ has to be a pedal point on $\omega(g_4, g_5, g_6)$ with respect to the point $(p_1, p_2, p_3) \in \mathbf{S}^2$ (see Fig. 3.2-right). Computationally, an easy way to obtain the closest pedal is to replace the underlying round metric by the Euclidean metric of the ambient space $\mathbb{R}^3$ and create the Lagrange function with an additional constraint, namely, the sphere equation. Note that by such an act one does not change the set $\mathscr{P}$ of pedal points on $\omega(p_4, p_5, p_6)$ with respect to (p_1, p_2, p_3). Therefore $\mathscr{P}$ can be computed as the variety $V(\frac{\partial L}{\partial u_1}, \frac{\partial L}{\partial u_2}, \frac{\partial L}{\partial u_3}, \frac{\partial L}{\partial \lambda_1}, \frac{\partial L}{\partial \lambda_2})$ where λ_1 and λ_2 are the Lagrange multipliers

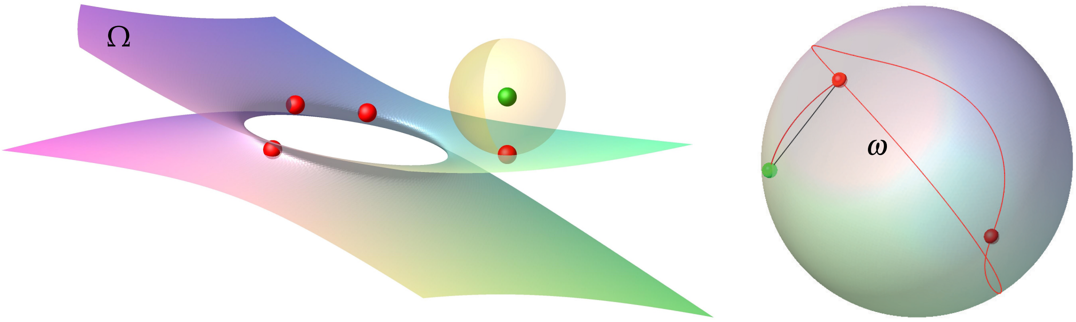


FIGURE 3.2: Illustrations are done for $\mathfrak{G} = (\frac{3}{5},\frac{4}{5},0,2,3,4)$ of the linear pentapod displayed in Fig. 2.1. Fixed orientation (Left): $\mathscr{O}$ has only four real solutions where the closest one $\mathfrak{O} = (\frac{3}{5},\frac{4}{5},2.4774,2.6978,0.9626)$ has a distance of 3.9444 units. Fixed position (right): $\mathscr{P}$ has only two real solutions where the closest one $\mathfrak{P} = (0.3701,0.5523,0.7468,2,3,4)$ has a remote distance of 48.4178°.

of the Lagrange equation

$$L(u_1,u_2,u_3,\lambda_1,\lambda_2) = (u_1-p_1)^2+(u_2-p_2)^2+(u_3-p_3)^2+\lambda_1 F+\lambda_2 G \tag{3.5}$$

with $G = u_1^2+u_2^2+u_3^2-1$ and F the singularity polynomial of Eq. 2.8.

Remark 13. *For the practical application of the round distance to the singularity, we recommend to locate the position vector* **p** *in the tool-center-point of ℓ.* ◇

The set $\mathscr{P}$ equals the variety of the ideal $\langle \frac{\partial L}{\partial u_1},\frac{\partial L}{\partial u_2},\frac{\partial L}{\partial u_3},\frac{\partial L}{\partial \lambda_1},\frac{\partial L}{\partial \lambda_2}\rangle$, which can be computed as follows:

Compute λ_1,λ_2 from the two equations $\frac{\partial L}{\partial u_1} = \frac{\partial L}{\partial u_2} = 0$, which are linear in u_1,u_2 and u_3. By plugging in the obtained expressions into $\frac{\partial L}{\partial u_3}$, $\frac{\partial L}{\partial \lambda_1}$ and $\frac{\partial L}{\partial \lambda_2}$ one gets three rational polynomials in the variables u_1, u_2 and u_3. We name their numerators F_1, F_2 and F_3, respectively. It turns out that these equations are quadratic. Since the solution set of these quadratic equations is $V(F_1,F_2,F_3) = V(F_1)\cap V(F_2)\cap V(F_3)$, the number of solutions is 8 according to *Bezout's Theorem*. Additionally, the number of terms of the polynomials F_1, F_2 and F_3 are 896, 348 and 4 respectively (F_3 is in fact the implicit equation of the sphere). Now in order to obtain these 8 solutions we use the resultant method in the following form:

$$R_1 := Res(F_2,F_3,u_1), \quad R_2 := Res(F_1,F_3,u_1), \quad R_3 := Res(F_1,F_2,u_1), \tag{3.6}$$

where R_1,R_2 and R_3 are dependent on the variables u_2 and u_3. By using the resultant method again to eliminate the variable u_2 we obtain

$$G_1 := Res(R_2,R_3,u_2), \quad G_2 := Res(R_1,R_3,u_2), \quad G_3 := Res(R_1,R_2,u_2). \tag{3.7}$$

The greatest common divisor of G_1,G_2,G_3 yields the degree 8 polynomial in u_3.

Example 21. *It turns out that for an example with the same architectural design as Eq. 3.4 (i.e. pose $\mathfrak{G}$ is given in the caption of Figs 3.2 - right) one obtains only 2 solutions are real (see Figs. 3.2 - right and 3.3 - left). The corresponding values of u_1,u_2,u_3,λ_2 are obtained by back-substitution (cf. Table 3.3).*

	u_1	u_2	u_3	λ_1	λ_2	s
1	0.3701	0.5523	0.7468	0.0000002748	4.3813	48.4178°
2	-0.3265	-0.5850	-0.7423	0.0000131667	-6.4345	131.6726°

TABLE 3.3: The 2 real solutions in ascending order with respect to the round distance s to the given orientation.

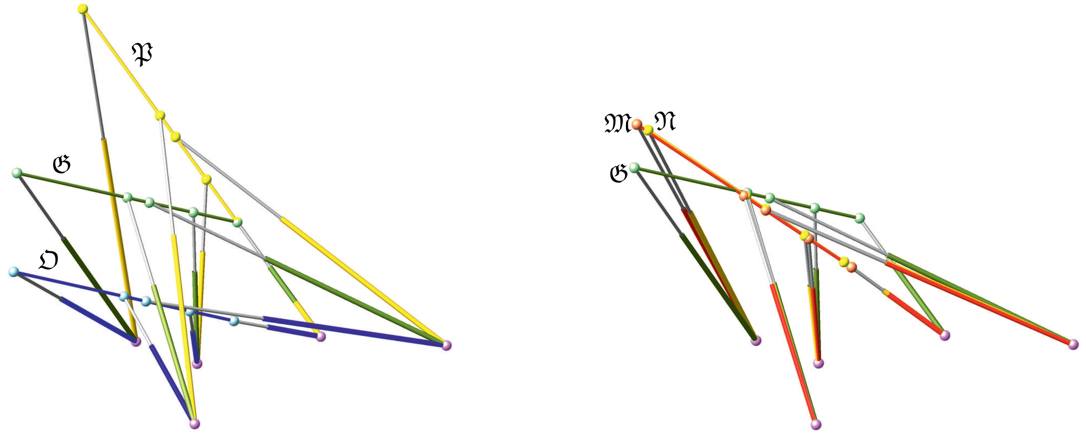


FIGURE 3.3: Given is the pose $\mathfrak{G}$ (black) of the linear pentapod. Left: The closest singular configurations in the position/orientation workspace are given by the pose $\mathfrak{P}$ (yellow) and $\mathfrak{O}$ (blue), respectively. Right: $\mathfrak{M}$ (red) is the closest singular pose under Euclidean motions of ℓ and $\mathfrak{N}$ (yellow) is the closest singularity under equiform motions of ℓ.

3.2.3 Euclidean Motion

In contrast to the two cases discussed before, Euclidean motion case deals with mixed (translational and rotational) DOFs. Therefore, in this case, one deals with the metric space $\mathbf{S}^2 \times \mathbb{R}^3$ associated with object-oriented metric (cf. Eq. 3.2 and Table 3.1). With respect to this metric $\mathfrak{d}$ we can compute the closest singular configuration $\mathfrak{M}$ to $\mathfrak{G}$ in the following way: Determine the set $\mathscr{M}$ of pedal points on the singularity variety with respect to $\mathfrak{G}$ as the variety $V(\frac{\partial L}{\partial u_1}, \frac{\partial L}{\partial u_2}, \cdots, \frac{\partial L}{\partial u_6}, \frac{\partial L}{\partial \lambda_1}, \frac{\partial L}{\partial \lambda_2})$ where λ_1 and λ_2 are the Lagrange multipliers of the Lagrange equation

$$L(u_1, u_2, \cdots, u_6, \lambda_1, \lambda_2) := \mathfrak{d}(\mathfrak{M}, \mathfrak{G})^2 + \lambda_1 G + \lambda_2 F. \tag{3.8}$$

Random examples indicate that $\mathscr{M}$ consists of eighty points over $\mathbb{C}$, where the one with the shortest distance d to $\mathfrak{G}$ equals $\mathfrak{M}$ (see Fig. 3.3 - right).

Remark 14. *Note that these minimal distances are seen as the radii of maximal singularity-free hyperspheres [Li, Gosselin, and Richard, 2007] in the position workspace (see also [Nag et al., 2018]), the orientation workspace (see also [Jiang and Gosselin, 2009]) and the complete configuration space. Moreover the distance $\mathfrak{d}(\mathfrak{M}, \mathfrak{G})$ to the singularity variety can also be interpreted as quality index thus it is an alternative to the value of F proposed in [Borràs et al., 2009]. An alternative "closeness index" for the pentapod with planar base can be based on the results of [Borràs et al., 2009] that in singularity configurations legs with zero length can be constructed by the method of singularity-invariant leg-replacement. Therefore one can take the length of the shortest leg of the two-parametric set of legs, which can be allocated.* ◇

Gröbner Base: It is possible to compute the Gröbner basis of the ideal

$$\langle \frac{\partial L}{\partial u_1}, \frac{\partial L}{\partial u_2}, \cdots, \frac{\partial L}{\partial u_6}, \frac{\partial L}{\partial \lambda_1}, \frac{\partial L}{\partial \lambda_2} \rangle \tag{3.9}$$

by Maple using the FGb package of Faugère [Faugère, 2010] for a random example (e.g. architectural parameters given in Eq. 3.4 and pose $\mathfrak{G}$ given in the caption of Fig. 3.2). By means of this package we can also compute the univariate polynomial P in u_1. The corresponding Maple pseudo-code reads as follows:

```
 with(FGb):
GB:=fgb_gbasis([seq(diff(L,u[ii]),ii=1..6),diff(L,lambda[1]),...
                         diff(L,lambda[2])],0,[],[seq(u[ii],ii=1..3),lambda[1],...
                         lambda[2],seq(u[ii],ii=4..6)]):
P:=fgb_gbasis_elim(GB,0,[u[2],u[3],lambda[2],lambda[1],u[4],u[5],u[6]],  ...
                            [u[1],u[2],u[3],lambda[2],lambda[1],u[4],u[5],u[6]]):
```

It can easily be checked that P is of degree 80 in u_1.

Resultant method: We are also able to compute this polynomial P by a stepwise elimination of unknowns based on resultant method executed by Maple. Details of this approach read as follows[1]:
We start by computing u_4, u_5, u_6 from the three equations $\frac{\partial L}{\partial u_4} = \frac{\partial L}{\partial u_5} = \frac{\partial L}{\partial u_6} = 0$, which are linear in u_4, u_5, u_6. Plugging in the obtained expressions into $\frac{\partial L}{\partial u_1}$ shows that its numerator only depends linearly on λ_1. From this condition we compute λ_1 and insert it into the equations $\frac{\partial L}{\partial u_2} = \frac{\partial L}{\partial u_3} = \frac{\partial L}{\partial \lambda_2} = 0$, which only depend on u_1, u_2, u_3, λ_2. The remaining equation $\frac{\partial L}{\partial \lambda_1} = 0$ equals $G = 0$ with $G = u_1^2 + u_2^2 + u_3^2 - 1$. Then we compute the following resultants:

$$H_1 := Res(G_{u_2}, G, u_3), \quad H_2 := Res(G_{u_3}, G, u_3), \quad H_3 := Res(G_{\lambda_2}, G, u_3),$$

where G_i with $i \in \{u_2, u_3, \lambda_2\}$ denotes the numerator of $\frac{\partial L}{\partial i}(u_1, u_2, u_3, \lambda_2)$. Note that G_{u_2} is of degree 8 in u_3 and that G_{u_3} and G_{λ_2} are both of degree 9 in u_3. Moreover we have

$$H_1[1230], \quad H_2[1271], \quad H_3[1252],$$

where the number in the brackets gives the number of terms. It should also be mentioned that H_1 and H_2 are polynomials of degree 14 with respect to λ_2 and that H_3 is of degree 12 in λ_2. Then we proceed by computing

$$K_1 := Res(H_2, H_3, \lambda_2), \quad K_2 := Res(H_1, H_3, \lambda_2), \quad K_3 := Res(H_1, H_2, \lambda_2).$$

K_1, K_2, K_3 have two common factors, which do not cause solutions as they imply zeros in the denominators of above arisen expressions. Beside these factors K_1, K_2, K_3 split up into

$$K_{1,1}[2016]K_{1,2}[11175], \quad K_{2,1}[1938]K_{2,2}[11097] \quad \text{and} \quad K_{3,1}[1653]K_{3,2}[11371],$$

respectively, where the long factors $K_{j,2}$ (for $j = 1, 2, 3$) are caused by the elimination process and do not contribute to the final solution. The factors $K_{1,1}$ and $K_{2,1}$ are of degree 62 in v and $K_{3,1}$ is of degree 56 in u_2. The greatest common divisor of $Res(K_{1,1}, K_{3,1}, u_2)$ and $Res(K_{2,1}, K_{3,1}, u_2)$ yields the univariate polynomial P in u_1 of degree 80.

Example 22. *The polynomial P (either obtained by Gröbner basis elimination techniques or by the resultant method) has to be solved numerically. It turns out that for the random example under consideration only* 16 *solutions are real and* 64 *solutions are complex.*[2] *By back-substitution into the equations obtained during the stepwise elimination based on resultant method, we get the values for u_1, u_2, u_3, λ_2 (cf. Table 3.4).*

[1] Degrees and lengths of the given polynomials and factors are given with respect to the architectural parameters given in Eq. 3.4 and pose $\mathfrak{G}$ given in the caption of Fig. 3.2.

[2] It is unknown if examples with 80 real solutions can exist.

	u_1	u_2	u_3	λ_2	$\mathfrak{d}$
1	0.5559273	0.7274604	0.4021767	0.0000977	1.47919239
2	0.7100848	0.6097073	-0.3521880	0.0112120	6.37008978
3	0.6707364	0.6608309	-0.3367715	0.0263760	6.39634868
4	0.9520812	0.2971145	-0.0725547	0.0518935	6.49493069
5	-0.4198912	-0.7478308	-0.5142376	0.0001295	6.52284048
6	0.6426323	0.5670451	-0.5152508	0.6390457	7.90108999
7	-0.9141441	0.2145188	-0.3439800	0.0175335	8.15356091
8	-0.6633066	-0.6523793	0.3666407	-0.0198075	9.07264206
9	-0.4968498	-0.8534603	-0.1573075	0.0025204	9.24410297
10	0.4561177	-0.7759229	0.4357753	-0.1291505	9.30816713
11	-0.6449198	-0.5507646	0.5298459	0.0516473	9.97032291
12	0.9794782	-0.1982747	0.0361857	0.0967973	10.0548807
13	-0.2161351	0.9213140	-0.3232119	0.0362890	13.7804945
14	0.1003162	-0.5648716	-0.8190583	0.0430362	37.6037440
15	0.8001243	0.1449203	0.5820644	-0.1991801	52.2930848
16	0.0428010	0.5580832	0.8286804	-0.0557997	65.2624252

TABLE 3.4: The 16 real solutions in ascending order with respect to the distance $\mathfrak{d}$ (given in Eq. (3.2)) from $\mathfrak{G}$. The corresponding values of missing variables u_4, u_5, u_6, λ_1 are obtained by substituting u_1, u_2, u_3, λ_2 into the expressions for u_4, u_5, u_6, λ_1. For the global minimizer (solution 1) these values are $u_4 = 2.296688437$, $u_5 = 3.479406728$, $u_6 = 1.835729103$ and $\lambda_1 = -4.720444174$.

3.2.4 Equiform Motions

We can simplify the problem by considering equiform transformations of the linear platform ℓ. This means that we can cancel the side condition $G = 1$. The computation can be done in a similar fashion to Example 22 with the sole difference that we set $\lambda_1 = 0$.

Example 23. *Random examples show that this reduced problem has only* 28 *solutions over* $\mathbb{C}$ *in the general case and it turns out that only* 6 *solutions are real, which are given in Table 3.5. Moreover the global minimizer* $\mathfrak{N}$ *is displayed in Fig. 3.3-right. Important for application is that* $\mathfrak{d}(\mathfrak{N}, \mathfrak{G}) \leq \mathfrak{d}(\mathfrak{M}, \mathfrak{G})$ *and therefore the value of* $\mathfrak{d}(\mathfrak{N}, \mathfrak{G})$ *gives us the radius of hypersphere, which is guaranteed singularity free.*

	u_1	u_2	u_3	λ_2	$\mathfrak{d}$	μ
1	0.5055836	0.6656442	0.3718172	0.0000990	1.4517670	0.9148471
2	0.6486166	0.5542384	-0.3254994	0.0118010	6.3636100	0.9131449
3	0.6200661	0.6110910	-0.3126505	0.0283247	6.3914193	0.9250214
4	0.8789381	0.2541301	-0.0569840	0.0548756	6.4897306	0.9167124
5	0.5220226	0.4591892	-0.4197574	9.9482986	7.8756112	0.8121322
6	0.6513903	-0.9060852	0.5534569	-0.1501405	9.2038614	1.2456382

TABLE 3.5: The 6 real solutions in ascending order with respect to the distance $\mathfrak{d}$ (given in Eq. 3.2) from $\mathfrak{G}$. The scaling factor of the corresponding equiform displacement of the platform is given by μ. The corresponding values of missing variables u_4, u_5, u_6 are obtained by substituting u_1, u_2, u_3, λ_2 into the expressions for u_4, u_5, u_6. For the global minimizer $\mathfrak{N}$ (solution 1) these values are $u_4 = 2.5031164070$, $u_5 = 3.7266491740$ and $u_6 = 1.9989579769$.

3.3 Simple Pentapods

In this section we compute *singularity-free zones* for linear pentapods with a simple singularity variety studied in Sections 2.3.1, 2.3.2 and 2.4.

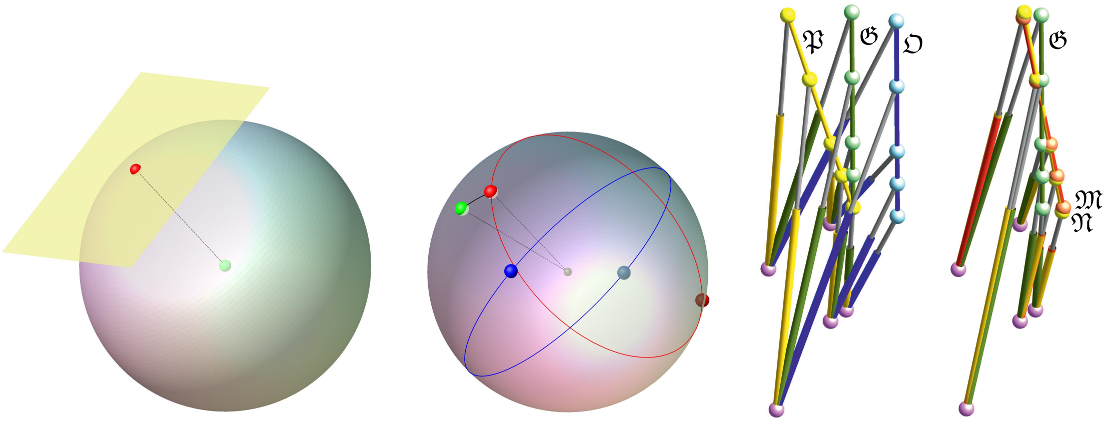


FIGURE 3.4: Left: For fixed orientation the unique pedal point has coordinates $(\frac{61}{33}, \frac{38}{33}, \frac{92}{33}) \in \mathbb{R}^3$ which has a distance of 1.21854359 units from the given position. Middle-left: For fixed position the four pedal points are illustrated, where the one with coordinates $(0.12661404, 0.81506780, 0.56536126) \in \mathbb{R}^3$ is closest to the given orientation. The corresponding spherical distance equals 15.75049156°. The second pedal point is antipodal to the first one and the spherical distance is the supplementary angle. Middle-right: Illustration of the pose $\mathfrak{G}$ (green) of the linear pentapod studied in Section 3.3.1. The closest singular configurations in the position/orientation workspace are given by the pose $\mathfrak{P}$ (yellow) and $\mathfrak{D}$ (blue), respectively. Right: $\mathfrak{M}$ (red) is the closest singular pose under Euclidean motions of ℓ and $\mathfrak{N}$ (yellow) is the closest singularity under equiform motions of ℓ.

3.3.1 Linear in Position Variables (LP)

The *architecture matrix* of the linear pentapod used in the examples of Section 3.3.1 is:

$$\begin{pmatrix} r_2 & x_2 & y_2 & z_2 \\ r_3 & x_3 & y_3 & z_3 \\ r_4 & x_4 & y_4 & z_4 \\ r_5 & x_5 & y_5 & z_5 \end{pmatrix} = \begin{pmatrix} 1 & -1/2 & 0 & 0 \\ 2 & 1 & 2 & 0 \\ 4 & -3 & -1 & 0 \\ 6 & -1 & 2 & 0 \end{pmatrix}, \tag{3.10}$$

where $\alpha = -2$ and $\beta = 2$ in Eq. (2.18). Moreover we consider the non-singular pose $\mathfrak{G} = (\frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 2, 3)$.

Positions (Fixed Orientation Case)

We ask for the closest singular configuration $\mathfrak{D}$ having the same orientation (g_1, g_2, g_3) as the given pose $\mathfrak{G}$. The distance to the singularity pose with respect to (g_4, g_5, g_6) is computed according to the ordinary *Euclidean* metric. The singularity polynomial is linear in position variables and under fixed orientation condition it will be a plane *passing through the origin* in position space $\mathbb{R}^3$. Naturally, there will be only one pedal point and hence the number of solutions in this case will only be one.

Example: A general example with $\mathfrak{D} = (\frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{61}{33}, \frac{38}{33}, \frac{92}{33})$ is illustrated in Fig. 3.4.

Orientations (Fixed Position Case)

Now we ask for the closest singular configuration $\mathfrak{P}$, which has the same position (g_4, g_5, g_6) as the given pose $\mathfrak{G}$. In this case the distance to the singularity curve with respect to (g_1, g_2, g_3) is computed according to the *Riemannian distance s* on the sphere. This means that the shortest path between two poses on the sphere is the shorter curve of the *great circle* connecting the two points.

Under the fixed position the singularity polynomial factors into two planes in $\mathbb{R}^3$:

$$u_3\,(A_1\,u_1 + A_2\,u_2 + A_3\,u_3 + A_4) = 0, \tag{3.11}$$

where the design variables are encoded in the coefficients A_i.

As a consequence the *singular orientations* are obtained as the intersection of these two planes with the unit-sphere, which is given by the normalizing condition $u_1^2 + u_2^2 + u_3^2 = 1$. One of these planes always passes through the center of the sphere and hence the intersection is a great circle. For the second plane different cases can occur:

(a) $A_4^2 < A_1^2 + A_2^2 + A_3^2$: the plane intersects the sphere.

(b) $A_4^2 = A_1^2 + A_2^2 + A_3^2$: the plane is tangent to the sphere.

(c) $A_4^2 > A_1^2 + A_2^2 + A_3^2$: the plane doesn't intersect the sphere.

Depending on the case the total number of pedal points equals (a) 4, (b) 3 and (c) 2, respectively.

Example: A general example with $\mathfrak{P} = (0.1266, 0.815, 0.5653, 1, 2, 3)$ is illustrated in Fig. 3.4. In the example at hand there exist 4 pedal points, which are listed in Table 3.6.

Remark 15. *It should be noted that if the given non-singular orientation is normal to one of the planes intersecting the unit-sphere, then there exists an infinite number of pedal points.* $\diamond$

	u_1	u_2	u_3	s
1	0.12661404	0.81506780	0.56536126	15.75049156°
2	0.44721359	0.89442719	0	41.83152170°
3	-0.44721359	-0.89442719	0	138.25977700°
4	-0.60029825	-0.34138359	-0.72325600	155.56475890°

TABLE 3.6: The 4 real solutions in ascending order with respect to the spherical distance s to the given orientation.

Euclidean Motion

With respect to the metric $\mathfrak{d}$ of Eq. (3.2), we can compute the closest singular configuration $\mathfrak{M}$ to $\mathfrak{G}$ in the following way:
We determine the set of pedal points on the singularity variety with respect to $\mathfrak{G}$ as the variety $V(\frac{\partial L}{\partial u_1}, \frac{\partial L}{\partial u_2}, \cdots, \frac{\partial L}{\partial u_6}, \frac{\partial L}{\partial \lambda_1}, \frac{\partial L}{\partial \lambda_2})$ where λ_1 and λ_2 are the *Lagrange multipliers* of the Lagrange equation:

$$L(u_1, u_2, \cdots, u_6, \lambda_1, \lambda_2) := \mathfrak{d}(\mathfrak{M}, \mathfrak{G})^2 + \lambda_1(u_1^2 + u_2^2 + u_3^2 - 1) + \lambda_2 F. \tag{3.12}$$

Note that here F is the singularity polynomial linear in position variables, obtained from Theorem 3.

Example: Considering the example of the design parameters indicated in Eq. 3.10, there are 10 solutions out of which 6 are real[3]. These solutions were calculated as follows:

After solving $\{\frac{\partial L}{\partial u_4}, \frac{\partial L}{\partial u_5}, \frac{\partial L}{\partial u_6}\}$ for $\{u_4, u_5, u_6\}$ and substituting the values obtained into the rest of the equations of the system, we can use the Gröbner basis method to solve the new system for the remaining variables. Using the order $u_3 > u_2 > u_1 > \lambda_2 > \lambda_1$ one of the

[3] It is unknown if examples with 10 real solutions can exist.

Gröbner basis generators solely depends on λ_1 while the rest depend on λ_1 and another orientation variable or λ_2, respectively. Based on this elimination technique the following table is obtained: The first row in Table 3.7 corresponds to the global minimizer $\mathfrak{M}$

	u_1	u_2	u_3	λ_1	λ_2	$\mathfrak{d}$
1	0.19954344	0.75426388	0.62551450	0.22471412	0.00242829	0.37163905
2	0.44721359	0.89442721	0.00000000	-1.18154819	0.15475648	1.53723662
3	-0.44720571	-0.89444123	0.00001503	-8.09845180	0.00888318	4.02454431
4	-0.72878205	-0.23306556	-0.64396839	-9.46430882	0.00812550	4.13597163
5	0.50116745	0.86532314	0.00686193	-1.24444052	63.53263267	4.98948239
6	-0.44100968	-0.89750916	-0.00554456	-8.10658006	-11.11676392	6.20308215

TABLE 3.7: The 6 real solutions in ascending order with respect to the distance $\mathfrak{d}$ from $\mathfrak{G}$. The corresponding values of missing variables u_4, u_5, u_6 are obtained by substituting $u_1, u_2, u_3, \lambda_1, \lambda_2$ into the expressions for u_4, u_5, u_6.

illustrated in Fig. 3.4, which has position variables $u_4 = 1.42386285$, $u_5 = 1.69623807$ and $u_6 = 3.11364494$.

Equiform Motion

We can simplify the problem by considering equiform transformations of the linear platform ℓ, which is equivalent to the cancellation of the normalizing condition Γ. This would be equal to canceling the first constraint in Eq. 3.12. It turns out that for this reduced set of equations only 3 pedal points exit over $\mathbb{C}$. For details see Theorem 9 given later on.

Example: For the example under consideration, the computations can be done in the same way as in Section 3.3.1 with the sole difference that λ_1 is now absent. We end up with the following table:

	u_1	u_2	u_3	λ_2	$\mathfrak{d}$	μ
1	0.22077150	0.77922849	0.65664594	0.00209764	0.35854952	1.04265095
2	0.33333333	0.66666666	0	0.04901408	1.43604394	0.74535599
3	0.36256185	0.63743814	0.01002046	26.26334956	4.95602764	0.73340227

TABLE 3.8: The 3 real solutions in ascending order with respect to the distance $\mathfrak{d}$ from $\mathfrak{G}$. The scaling factor of the corresponding equiform displacement of the platform is given by μ. The corresponding values of missing variables u_4, u_5, u_6 are obtained by substituting u_1, u_2, u_3, λ_2 into the expressions for u_4, u_5, u_6.

The first row in Table 3.8 corresponds to the global minimizer $\mathfrak{N}$ illustrated in Fig. 3.4, which has position variables $u_4 = 1.36501824$, $u_5 = 1.63498176$ and $u_6 = 3.03249538$.

3.3.2 Linear in Orientation Variables (LO)

The *architecture matrix* of the linear pentapod used in the examples of Section 3.3.2 is:

$$\begin{pmatrix} r_2 & x_2 & y_2 & z_2 \\ r_3 & x_3 & y_3 & z_3 \\ r_4 & x_4 & y_4 & z_4 \\ r_5 & x_5 & y_5 & z_5 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 3 & -1/2 & 3/2 & 0 \\ 5 & -3 & 4 & 0 \\ 6 & -1 & 2 & 0 \end{pmatrix}, \tag{3.13}$$

where $\alpha = \beta = 1$ in Eq. 2.29. Moreover we consider the non-singular pose is given as $\mathfrak{G} = (1/3, 2/3, 2/3, 1, 2, 3)$.

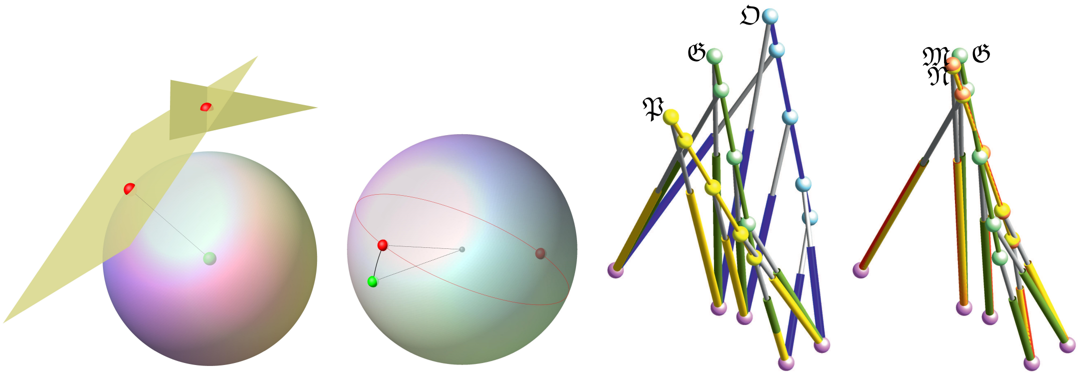


FIGURE 3.5: Left: For fixed orientation there exits for each of the two planes a unique pedal point. One pedal point has coordinates $(-8/17, 9/17, 12/17)$ and a distance of 4.80196038 units to the given position and the other pedal point has coordinates $(2,3,0)$ and a distance of 4 units. Middle-left: For fixed position the two pedal points are illustrated, where the one with coordinate $(0.11346545, 0.47007115, 0.87530491)$ is closest to the given orientation. The corresponding spherical distance equals 20.82450533°. Middle-right: Illustration of the pose $\mathfrak{G}$ (green) of the linear pentapod studied in Section 3.3.2. The closest singular configurations in the position/orientation workspace are given by the pose $\mathfrak{P}$ (yellow) and $\mathfrak{O}$ (blue), respectively. Right: $\mathfrak{M}$ (red) is the closest singular pose under Euclidean motions of ℓ and $\mathfrak{N}$ (yellow) is the closest singularity under equiform motions of ℓ.

Positions (Fixed Orientation Case)

Once again we ask for the closest singular configuration $\mathfrak{O}$ having the same orientation (g_1, g_2, g_3) as the given pose $\mathfrak{G}$. The distance to the singularity pose with respect to (g_4, g_5, g_6) is computed according to the *ordinary Euclidean* metric. Under fixed orientation condition it is revealed that the singularity polynomial is factored to:

$$u_6\,(B_1 u_4 + B_2 u_5 + B_3 u_6 + B_4) = 0, \tag{3.14}$$

where again the design information is encoded in coefficients B_i. For each of the two planes in position space $\mathbb{R}^3$ we can compute the pedal point with respect to the given pose (cf. Fig. 3.5).

Example: The closer pedal point implying $\mathfrak{O} = (\frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 2, 3, 0)$ is illustrated in Fig. 3.5.

Orientations (Fixed Position Case)

Now we ask again for the closest singular configuration $\mathfrak{P}$, which has the same position (g_4, g_5, g_6) as the given pose $\mathfrak{G}$. As the singularity polynomial is linear in orientation variables and does not possess an absolute term, the *singularity loci* is a *great circle* for the fixed position case. If the given orientation differs from the pole of the great circle, then there exist two pedal points (otherwise infinitely many).

Example: The results for the example at hand and the pose

$$\mathfrak{P} = (0.1134, 0.4700, 0.8753, 1, 2, 3)$$

are displayed in Fig. 3.5.

Euclidean Motions

Similar “experimental" computations as in Section 3.3.1 show that there are again 10 solutions.

Example: For the example at hand we obtain 6 real solutions, which are given in Table 3.9:

	u_1	u_2	u_3	λ_1	λ_2	$\mathfrak{d}$
1	0.24002202	0.57831003	0.77970951	-0.07616071	0.00198708	0.41484860
2	0.16067752	0.32134537	0.93924532	5.58789193	-0.03317073	2.44661840
3	-0.20843306	-0.55064498	-0.80863487	-10.27182281	0.00088059	4.53615852
4	-0.35275218	0.88481355	-0.34421986	-3.79940039	0.38394736	6.70384275
5	-0.02624291	-0.92437183	0.38072309	-7.13002767	0.19314992	7.16835476
6	-0.06268654	-0.12537309	-0.99012725	-32.85080126	0.07233642	9.04867032

TABLE 3.9: The 6 real solutions in ascending order with respect to the distance $\mathfrak{d}$ from $\mathfrak{G}$.

The first row in Table 3.9 corresponds to the global minimizer $\mathfrak{M}$ illustrated in Fig. 3.5, which has position variables $u_4 = 1.35978906$, $u_5 = 2.34492506$ and $u_6 = 2.57706069$.

Equiform Motions

Similar computations as in Section 3.3.1 show again that the number of solution reduces to three (this is discussed in the Theorem 9 below).

Example: For the example at hand all three are real and read as follows: The first row in

	u_1	u_2	u_3	λ_2	$\mathfrak{d}$	μ
1	0.23632218	0.56965551	0.76841946	0.00196374	0.41349741	0.98530404
2	0.33333333	0.66666666	1.30046948	-0.02111913	1.81542685	1.49892509
3	-0.06965551	0.26367781	-0.10175277	3.26647730	6.49924087	0.29108677

TABLE 3.10: The 3 real solutions in ascending order with respect to the distance $\mathfrak{d}$ from $\mathfrak{G}$.

Table 3.10 corresponds to the global minimizer $\mathfrak{N}$ illustrated in Fig. 3.5, which has position variables $u_4 = 1.36986410$, $u_5 = 2.36986410$ and $u_6 = 2.61205791$.

Theorem 9. *Consider an arbitrary point p in $\mathbb{R}^6$. Then with respect to the object-oriented metric, Eq. 1.12, the corresponding number of pedal points on the Σ-variety of a simple pentapod is up to 3 and the pedal point coordinates can be obtained in a closed form.*

Proof. In Theorem 8, it is shown that the Σ-variety decomposes into the hyperplane Σ_1 and the hyperquadric Σ_2. Trivially, one of the pedal points is located on Σ_1, namely, the *closest* point (with respect to the distance function) to the arbitrary point p. Now, knowing this fact it is possible to focus on Σ_2 for retrieving the pedal points information. Consider the following *Lagrange equation*

$$L := \mathfrak{d}^2\,(p,u) + \lambda\, F \tag{3.15}$$

where λ is the *Lagrange multiplier*, F is the singularity polynomial of Σ_2 and $\mathfrak{d}$ is the *object-oriented metric* describing the distance between the point p and a sample point u on Σ_2 with symbolic coordinates $(u_1, u_2, \ldots, u_6)$. Then, computing

$$I := \langle \frac{\partial L}{\partial u_1}, \frac{\partial L}{\partial u_2}, \ldots, \frac{\partial L}{\partial u_6}, \frac{\partial L}{\partial \lambda} \rangle \tag{3.16}$$

	Number of Pedal Points		
Motion	Generic	LP	LO
Translations	8	1	2
Orientations	6	4	2
Euclidean motion	80★	10★	10★
Equiform motion	28★	3	3

TABLE 3.11: Illustration of the generic number of pedal points under different metric conditions. LP/LO terms stand as linear in position/orientation variables while “generic case" refers to the general pentapod (not necessarily with a simple singularity variety). Note that they are computed over the field of complex numbers and hence the real solutions might be lower. The entries marked with ★ indicate that these numbers are just experimental while the rests are mathematically proven and reliable in all situations.

results in a system of polynomial equations, where $\partial L/\partial u_1,\ldots,\partial L/\partial u_6$ are linear in the pose variables. By solving them for $u_1,\ldots,u_6$ and substituting them in $\partial L/\partial\lambda$ one obtains a univariable polynomial quadratic in λ. Since this polynomial is of degree 2 it shows that for the arbitrary point p there will be at most 2 pedal points on Σ_2. Finally, the pedal point coordinates are obtained through solving the quadratic polynomial for λ and back-substituting its value. □

3.3.3 Summary of Results

We demonstrated in this section that these designs imply a degree reduction of the polynomials associated with the problem of determining singularity-free zones. One should note that the results (cf. Table 3.11) obtained for singularity-free zones regarding manipulators with singularity varieties linear in position/orientation variables in the cases of *fixed position variables* and *fixed orientation variables* are general while for the *general case with normalizing condition* the given numbers are just based on a set of random examples[4]. Theorem 9 shows that the pedal point problem in the *equiform motions* case is a cubic one. Therefore the closest singular configurations under equiform motions (cf. Sections 3.3.1 and 3.3.2) are of interest, as they can be computed in closed form. This property will also be stressed in the next section.

[4] In this context we also refer to Section 6 “What is a proof?" of [Faugère and Lazard, 1995], where the authors Faugere and Lazard were faced with a similar problem.

Chapter 4

Variational Path Optimization of Simple Pentapods

4.1 Preparatory Work

Assume that a path between a given start- and end-pose of the end-effector is known, which is singularity-free and within the manipulator's workspace. An optimization process of this initial path is proposed in such a way that the parallel robot increases its distance to the singularity loci while the motion is being smoothed. In our case the computation time of the optimization is improved as we are dealing with pentapods having simple singularity varieties allowing a closed form solution for the local extrema of the singularity-distance function. Formally, this process is called *variational path optimization* which is the systematic optimization of a path by manipulating its *variations of energy* and *distance to the obstacle*, which in this case is the singularity variety. In this process some physical limits of the mechanical joints are also taken into account. Furthermore, for the sake of simplicity, from time to time, the theoretical aspects of the current chapter are accompanied by examples in $\mathbb{R}^2$.
The rest of the chapter is organized as follows: First the concept of distance to the singularity loci of a simple pentapod is investigated. Thanks to the results of Chapter 3, we now have a thorough understanding of the possible outcomes. However, there are still some details to be investigated further, namely, the pedal points on the *singular points of singularity loci* and pose points that are located on *cut locus of singularity loci*. On the next pace, a *cost function* is defined. This cost function gives us a *global control* over the motion optimization. This function is designed in such a way to control the smoothness of the motion while keeping the motion at a reasonable distance to the singularity loci. Furthermore, the conditions under which the motion is kept within the joint rate limits is studied. These include the case of prismatic and base spherical joints. Finally, the algorithm's steps are studied in detail and some examples are illustrated. These examples are created in such a way to cover general case but also to contain some possible emerging *extreme* cases.

4.1.1 Distance to Σ-variety & Modified Orthogonal Projection

Obtaining the closed form pedal point coordinates, enables one to compute the vector field of normals to Σ-variety at these points. Assuming a pose point $p \in \mathbb{R}^6$, the *natural* way of increasing distance to the Σ-variety is to move on the line spanned by p and the closest pedal (from now on called *normal lines*). However, this natural primary idea needs to be modified as the following cases may occur:

Pedal points on Σ_3

Despite the fact that, based on Definition 4, there is no restriction for the closest point to any specific subset of the Σ-variety, if such a point is located on Σ_3, Definition 9 does not reveal

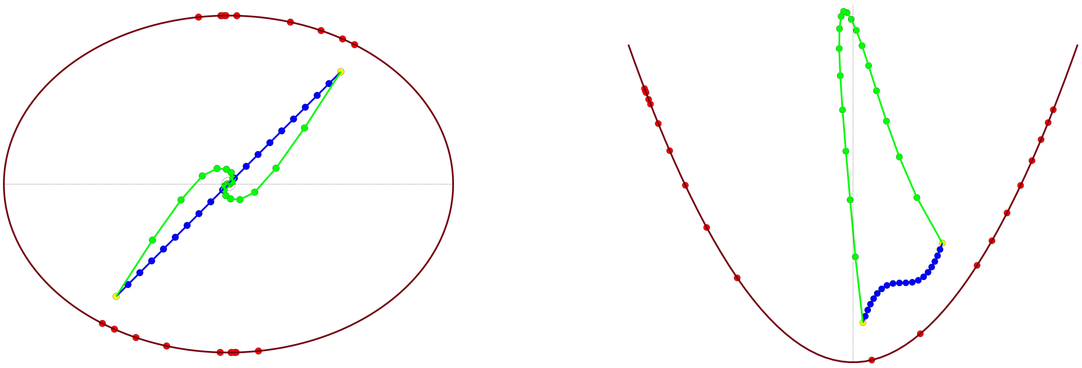


FIGURE 4.1: Illustrations of overshooting over the cut locus. On the left, if the points of an initial curve (blue) tend to increase their distance to the ellipse by travelling along the *normal lines* it is expected to witness the breakpoints to first land on the medial axis and then all pile up at the center of the ellipse. But in practice, due to the *jump* of the corresponding pedal point, an overshooting takes place. On the right, it is expected to see the breakpoints to land on the medial axis and slowly go to infinity. However, in practice, the break points oscillate horizontally and slowly move vertically.

it. The reason behind it is that the Lagrange multiplier technique nourishes from the tangency of the *distance function* and the *constraint* at the pedals. In another word, such minimizer is found whenever the corresponding gradients of the distance function and the constraint are linearly dependent. On the other hand, at singular points of the constraint the gradient vanishes which results the incapability of the Lagrange multipliers in identifying pedals on Σ_3 (cf. Chapter 1 and Fig. 1.11). One might try to overcome this hardship through redefining the pedal points as the solutions to the system of polynomials obtained by calculating the derivatives of the *distance function* with respect to parameters of Σ_2. In such a method one obtains the pedals on Σ_3 at the cost of expensive symbolic computation of non-linear systems. Though this technique is not reasonable, resolving this issue is not hard as one can find the closest point on Σ_3 separately. Hence, from now on we loosely call this point a pedal point as well (for details of Σ_3 cf. Theorem 8). Note that in the light of Lemma 1 we can still call the connecting line between the arbitrary point and its corresponding pedal point on Σ_3, a *normal line*.

Pose points on cut locus

By Theorem 9 and *previous case*, we know that p can be located on up to four *normal lines*. A natural question would be which of these line to travel along with to increase the distance to Σ. One idea would be to consider taxing along the normal line at the *closest* pedal. But if the pose point is located on the *cut locus* of Σ then more than one closest point already exist (for definition and properties of cut locus see Section 1.1.6 and [Carmo, 1992]). On the other hand, as pose point updating is discrete, with respect to a step size, it may create problems such as overshooting the cut locus. Such overshooting happens due to a *jump* of the corresponding closest pedal (see Fig. 4.1). In order to resolve both issues at once and create a smooth update of the pose point, a *weight* factor is introduced in such a way that it considers the effect of the pedal points in dependence of their distances to the pose p.

Modified orthogonal projection

Due to the reasons mentioned above, we modify Definition 4 into the following shape:

Definition 30. *The orthogonal projection with respect to object-oriented metric is a relation $\pi \subset \Sigma \times \mathbb{R}^6$, such that for each $p \in \mathbb{R}^6$ we have*

$$\pi(p) := \{\textit{closest point on } \Sigma_1\} \cup \mathbf{V}(I) \cup \{\textit{closest point on } \Sigma_3\}, \tag{4.1}$$

where I is Eq. 3.16 (cf. Theorem 9). From now on through the entire chapter, π refers to the "modified orthogonal projection".

Note that in a generic case, $\pi(p)$ is finite and yields four pedal points (one on Σ_1, two on Σ_2 and one on Σ_3).

Remark 16. *Note that, it is not immediately known whether the non-real complex pedal points or double pedal points exist or not. Calculating Eq. 3.16 for the case of a* parabola *as an obstacle shows that the pedal points are not always distinct real points.*

Definition 31. *Assume $p \in \mathbb{R}^6$ and π to be an orthogonal projection relation mentioned above. Additionally, assume m_p to be the number of the real pedal points on Σ with respect to the point p. Now, the goal is to find a preferred direction such that if point p travels along with, it can increase its distance to the Σ-variety. Assuming u to be the update of p, the following function fulfills the goal:*

$$\mathscr{D}(p,u) := -\sum_{i=1}^{m_p} w_i \langle \frac{p-\pi_i(p)}{\|p-\pi_i(p)\|_g}, u-p \rangle, \tag{4.2}$$

where the $w_i := H/d(p,q_i)$ are the "positive weights" and H given by

$$H := \frac{\prod_{i=1}^{m_p} \mathfrak{d}(p,q_i)}{\sum_{j=1}^{m_p} \prod_{k \neq j} \mathfrak{d}(p,q_k)} \tag{4.3}$$

is the "regulation factor" implying $\sum_{i=1}^{m_p} w_i = 1$.

Eq. 4.2 refers to a weighted projection of the vector $\overrightarrow{pu}$ on the four *normal lines*. As it can be inferred from Eq. 4.3, these weights are disproportional to the corresponding distance to pedals and their sum is equal to one. In another word, if p is far away from a pedal then the projection of movement direction, $\overrightarrow{pu}$, on that normal is a smaller vector (going away from that pedal is less important). On the contrary, if p is close to a pedal then the projection of $\overrightarrow{pu}$ would be mainly along that normal line (it will repel the point p with more efficiency). In Fig. 4.2-left an example regarding such weighted factors is depicted.

4.1.2 Cost Function

In order to find the desired *nice motion* the idea is to use *gradient descent*, which is a first-order iterative optimization algorithm, to find the local minimum of a *cost function*. Though in many similar applications, especially in the context of machine learning, the step size (learning rate) is considered a constant small number, here it will not be accurate enough and hence a more intelligent step should be taken for that purpose. Additionally, the function whose descent per iterations is monitored is slightly different from the original cost function and will be introduced under the name of *objective function* (cf. Section 4.1.4).

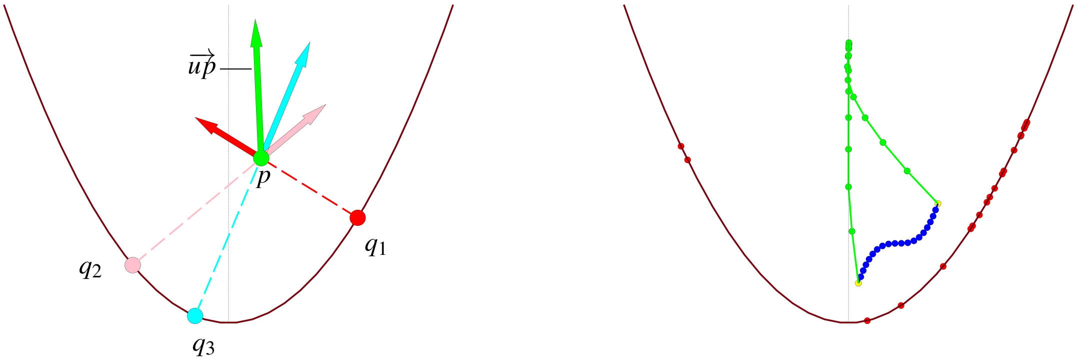


FIGURE 4.2: Left: The green vector shows the optimized direction for the vector $u-p$. The projection on the 1st, 2nd and 3rd gradient lines are indicated by red, pink and cyan respectively. Note that, one can check the correctness of the optimized move as the green direction refers to moving closer to the medial axis and increasing distance to the parabola eventually. Right: Depending on the geometrical structure of the obstacle, geodesic and bending weight factors, the algorithm may not have a final result. The initial path (blue) variates (green) along the parabola's medial axis till infinity under zero geodesic and bending weight inputs.

Definition 32. *Assume that an initial singularity-free curve between start-pose a' and end-pose b' is given, while it is discretized into a "piecewise smooth" curve with a set of n break-points $\{a' = p^1, p^2, ..., p^n = b'\}$ in $\mathbb{R}^6$. Assuming u^j to be the update of p^j, the cost function, subject to optimization, is defined as follows:*

$$\mathscr{C}(p,u) := \frac{\lambda\,(n-1)}{2L'}E(u) + \frac{\eta\,(n-2)}{2\,\tau'}B(u) - \frac{1}{(n-2)}\sum_{j=2}^{n-1}\mathscr{D}\left(p^j,u^j\right), \tag{4.4}$$

where $\mathscr{D}$ is the function of Eq. 4.2, λ and η are real numbers called "geodesic weight" and "bending weight", respectively and L' and τ' are the discrete length and discrete total curvature obtained from previous iteration (at the first iteration they are substituted by the corresponding values of the initial curve). Finally, we have $u^T := \left((u^2)^T, ..., (u^{n-1})^T\right)$ and $p^T := \left((p^2)^T, ..., (p^{n-1})^T\right)$.

Geodesic and bending weights increase/decrease the geodesic energy and bending energy by blocking/unblocking their increase when substituted by high/low values.

Remark 17. *In the absence of the coefficients $1/L'$ and $1/\tau'$ in Eq. 4.4, the geodesic energy and bending energy terms which are the sum of the "squared values" would heavily outgrow the distance term containing the projection on the normal lines which is single valued (cf. Fig. 4.3). This fact forces the cost function to be heavily dependant on the number of chosen breakpoints. Consequently, the existence of these coefficients are necessary.*

Remark 18. *It is noteworthy that the coefficients of the geodesic and bending energy terms in Eq. 4.4 are multiplied by a factor involving the number of breakpoints while the projection term's coefficient plays the role of a mean value for the number of breakpoints. These coefficients are designed in this way to reduce the effect of the number of breakpoints on the shape of the optimized curve. This is important as later it is shown that the number of break points may vary at different iterations of the optimization algorithm.*

The cost function, $\mathscr{C}$, is a *quadratic* polynomial in $6\,(n-2)$ variables. Optimizing Eq. 4.4 requires solving the following equation

$$\nabla\mathscr{C} = 0, \tag{4.5}$$

for $u \in \mathbb{R}^{6\,(n-2)}$ which is equivalent to solving the linear system of equations

$$\mathbf{A}\,\mathbf{u} = \mathbf{b}, \tag{4.6}$$

Where $\mathbf{A}$ is the $6\,(n-2) \times 6\,(n-2)$ matrix of coefficients, $\mathbf{u}$ is the $6\,(n-2) \times 1$ matrix of variables and $\mathbf{b}$ is the $6\,(n-2) \times 1$ matrix of constants. The computational details of these matrices and the patterns they obey are given in Appendix C.

4.1.3 Step Size

The idea is to calculate two step sizes, namely γ_1 and γ_2, in such a way that they yield $\pm\text{growth}\,\%$ of variations of geodesic and bending energy respectively. In doing so, the cost function, which can be seen as a quadratic approximation of the objective function yielding a linear solution for the gradient descent approach, is leashed to the objective function in a computational efficient way, as γ_1 and γ_2 can be computed in closed form as follows.

$$P(\gamma_1) := \sum_{i=2}^{n} \|\left(p^i + \gamma_1\,u^i\right) - \left(p^{i-1} + \gamma_1\,u^{i-1}\right)\|_g^2, \tag{4.7}$$

and its bending energy by

$$Q(\gamma_2) := \sum_{j=2}^{n-1} \|\left(p^{j-1} + \gamma_2\,u^{j-1}\right) + \left(p^{j+1} + \gamma_2\,u^{j+1}\right) - 2\,\left(p^j + \gamma_2\,u^j\right)\|_g^2, \tag{4.8}$$

and having the previous geodesic energy E' and previous bending energy B', then the real roots of the following quadratic equations will guarantee $\pm\text{growth}\%$ variation in energies:

$$P(\gamma_1) - (1 \pm \frac{\text{growth}}{100})E' = 0, \quad Q(\gamma_2) - (1 \pm \frac{\text{growth}}{100})B' = 0. \tag{4.9}$$

Finally, the step size is chosen as the minimum of the positive real roots of Eq. 4.9 and 1.

4.1.4 Objective Function

Now, in order to monitor the descent of the cost function with respect to iteration we propose a slightly different function called *objective function*.

Definition 33. *Assume u to be the vector obtained by solving the linear system (cf. Section 4.1.2) and π the orthogonal projection into the Σ-variety. Then the “objective function" for monitoring the gradient descent is*

$$\mathscr{C}'\,(u,p) := \frac{\lambda\,(n-1)}{2L}\,E\,(u) + \frac{\eta\,(n-2)}{2\,\tau}\,B\,(u) - \frac{1}{(n-2)}\sum_{j=2}^{n-1}\mathfrak{d}\left(u^j,\pi\left(p^j\right)\right). \tag{4.10}$$

Note that, as we have the updated curve breakpoint coordinates, it is possible to compute length and total curvature now. Hence, the terms $1/L$ and $1/\tau$ replace the terms $1/L'$ and $1/\tau'$ from Eq. 4.4 respectively. Additionally, since the goal is to increase the distance to the Σ-variety, the corresponding terms are replaced by the sum of distances to corresponding closest pedals.

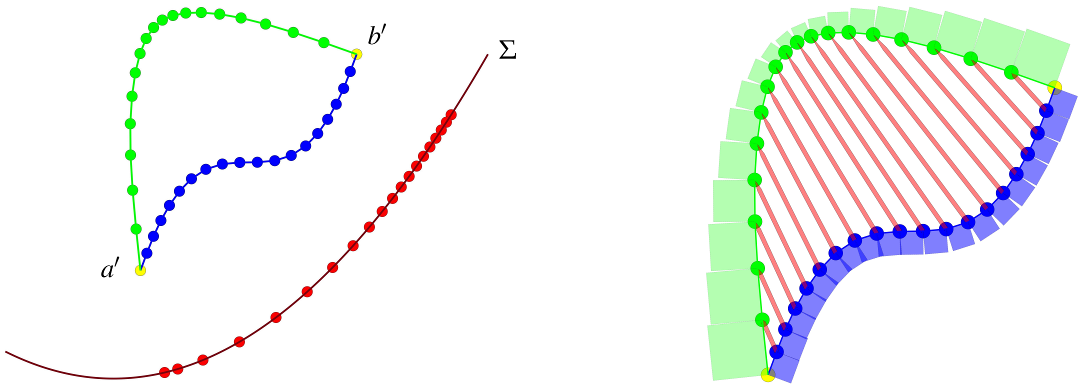


FIGURE 4.3: Left: Illustration of the pull effect of geodesic energy on the breakpoints. The pattern of the breakpoints preserves a regular shape after variation. Right: A geometrical visualization of the errors. The blue boxes describe the calculated value for the geodesic energy before variation, while the green ones depict this energy after the variation. The red arrows connect a breakpoint to its corresponding update. One way to realize the error's occurrence would be through assigning physical units to the calculations, namely *cm*. In that way the the value computed for the geodesic energy will be of physical dimension cm^2 while the distance is *cm*. Both figures are created for the case of the variation of an initial curve $t \mapsto \left(t+1.25, t^3+2\right)$ where $t \in [-1,1]$ with respect to a parabola and with $\lambda = \eta = 10$ as weight factors.

4.1.5 Remarks on the Curve's Energies

In the light of Definition 10, one can define the bending energy by $\sum_{i=1}^{n} \theta_i^2$, but Definition 12 is slightly different. Due to the fact that the measure of a central angle and the arc it intercepts are equal in value, in the case that the curve is parametrized by arc length the two definitions are equivalent. However, such values obtained may differ if the curve under consideration is not arc length parametrized (i.e. consider a line not possessing arc length parametrization, then based on Definition 12 the bending energy is not zero). But such a deviation creates a computational advantage as otherwise a denominator containing the norms of every two adjacent edges would appear in the bending energy term B of the cost function (Eq. 4.4). This denominator then diminishes bending energy term being a quadratic polynomial and consequently denying the advantage of possessing a linear system after derivation. Finally, the geodesic energy term contributes significantly to resolving this issue as it creates a *pull* effect which preserves a regular distribution of breakpoints across the curve after variation (see Fig. 4.3 - right).

4.2 Orthogonal Projection into the Configuration Space

In the process of updating breakpoints, by obtaining the preferred direction and step size (cf. Eq. 4.9), one may end up with a curve not necessarily belonging to the cylinder Γ. Having in mind that Γ is the set of all performable motions for the pentapod, it is important to find an update restricted to it. In order to resolve it, one immediate idea is to keep the update of each breakpoint p restricted to its corresponding tangent space $T_p(\Gamma)$ of the cylinder Γ. In the next step, the updated curve is orthogonally projected into the cylinder.

4.2.1 Projection on Cylinder's Tangents

For a breakpoint p we are interested in its update u belonging to $T_p(\Gamma)$. The Lemma 2 creates the first step in building a systematic update of $\left\{p^2, \ldots, p^{n-1}\right\}$ on $T\Gamma$.

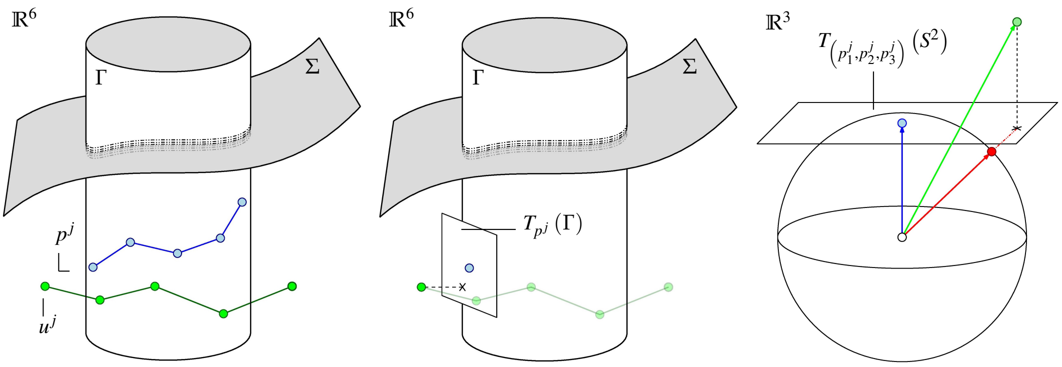


FIGURE 4.4: Imaginative illustration of the “generalized cylinder" $\Gamma \subset \mathbb{R}^6$. Left: The blue curve indicates the initial path while the green one stands as the updated curve (not necessarily belonging to Γ). The red dots are in fact the possible imaginative pedal points on the singularity surface Σ. Middle: The projection of the updated curve into the the tangent space. Right: A “realistic" interpretation of such projections in the orientation space. The blue vector shows the original orientation of the pose p^j and the green vector depicts the orientation after update. The cross, x, stands as the back-projection into the “tangent to sphere at p^j", while the red vector describes the final accepted motion after back-projection from the tangent to sphere.

Each breakpoint of the piecewise smooth curve is located in the ambient space $\mathbb{R}^6$. Define

$$\hat{p} = \left(\ (p\,{}^{2}_{6\times1})^T, (p\,{}^{3}_{6\times1})^T, \cdots, (p\,{}^{n-2}_{6\times1})^T \ \right)^T_{6(n-2)\times1}, \tag{4.11}$$

where $p\,{}^{i}_{6\times1}$ denotes the 6-dimensional coordinate vector of the i-th breakpoint of the piecewise smooth curve. In this way the point $\hat{p} \in \mathbb{R}^{6(n-2)}$ stands for the piecewise smooth curve. Defining the updated breakpoint u in exactly the same fashion we have

$$\hat{u} = \left(\ (u\,{}^{2}_{6\times1})^T, (u\,{}^{3}_{6\times1})^T, \cdots, (u\,{}^{n-1}_{6\times1})^T \ \right)^T_{6(n-2)\times1}. \tag{4.12}$$

Now, using the above terminology one trivially finds that the vector $\hat{u}-\hat{p}$ is in fact the $(\nabla\mathscr{C})_{\hat{p}}$ on the level sets of the cost function in $\mathbb{R}^{6(n-2)}$.

Lemma 10. *Assume $\hat{p} \in \Gamma^{n-2} \subset \mathbb{R}^{6(n-2)}$. Then*

$$\left(\nabla\mathscr{C}|_{\Gamma^{(n-2)}}\right)_{\hat{p}} = \left(\ (\mathscr{U}\,{}^{2}_{6\times1})^T, (\mathscr{U}\,{}^{3}_{6\times1})^T, \cdots, (\mathscr{U}\,{}^{n-1}_{6\times1})^T \ \right)^T_{6(n-2)\times1} - \hat{p}, \tag{4.13}$$

where $\forall i,\ 2 \le i \le n-1$ we have

$$(\mathscr{U}\,{}^{i}_{6\times1})^T = \left(\mathrm{Pr}_{(p_1^i,p_2^i,p_3^i)}\left(u_1^i,u_2^i,u_3^i\right), u_4^i, u_5^i, u_6^i\right)^T, \tag{4.14}$$

where $\mathrm{Pr}_{(p_1^i,p_2^i,p_3^i)} : \mathbb{R}^3 \longrightarrow T_{(p_1^i,p_2^i,p_3^i)}\left(S^2\right)$ is the orthogonal projection onto the sphere's tangent plane at $\left(p_1^i,p_2^i,p_3^i\right)$.

Proof. Γ is a regular submanifold of $\mathbb{R}^6$. This implies Γ^{n-2} being a regular submanifold of $\mathbb{R}^{6(n-2)}$. Now, using Lemma 2 we have $\left(\nabla\mathscr{C}|_{\Gamma^{(n-2)}}\right)_{\hat{p}} = \mathrm{Pr}_{\hat{p}}\left(\nabla\mathscr{C}\right)_{\hat{p}} = \mathrm{Pr}_{\hat{p}}\left(\hat{u}-\hat{p}\right)$ where $\mathrm{Pr}_{\hat{p}} : \mathbb{R}^{6(n-2)} \longrightarrow T_{\hat{p}}\left(\Gamma^{n-2}\right)$ is the orthogonal projection into the tangent of Γ^{n-2} at $\hat{p}$. This results in

$$\left(\ \left(\mathrm{Pr}_{p^2}\left(u^2\right)\right)^T, \left(\mathrm{Pr}_{p^3}\left(u^3\right)\right)^T, \cdots, \left(\mathrm{Pr}_{p^{n-1}}\left(u^{n-1}\right)\right)^T \ \right)^T_{6(n-2)\times1} - \hat{p}, \tag{4.15}$$

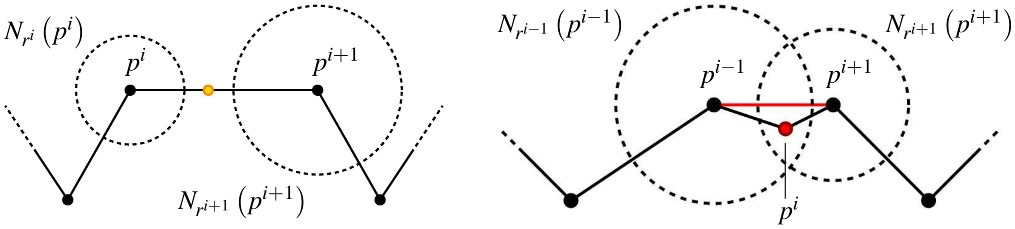


FIGURE 4.5: Imaginative illustration of minimal singularity-free cover of the curve. Left: If a segment is not fully covered by singularity-free balls then a breakpoint will be added in the mid of the uncovered part. Right: If two adjacent singularity-free balls are covering a breakpoint that breakpoint is removable. Note that such a removal should be in such a way that it does not alter the cover for adjacent to the neighboring breakpoints.

where $\mathrm{Pr}_{p^i} : \mathbb{R}^6 \longrightarrow T_{p^i}(\Gamma)$ is the orthogonal projection onto the tangent space at $p^i \in \Gamma$. Now having in mind that:

$$T_{\hat{p}}\left(\Gamma^{n-2}\right) \cong T_{p^2}(\Gamma) \times T_{p^3}(\Gamma) \times \ldots T_{p^{n-1}}(\Gamma) \cong \tag{4.16}$$

$$T_{(p_1^2,p_2^2,p_3^2)}\left(S^2\right) \times T_{(p_1^3,p_2^3,p_3^3)}\left(S^2\right) \times \ldots T_{(p_1^{n-1},p_2^{n-1},p_3^{n-1})}\left(S^2\right) \times \mathbb{R}^{3(n-2)}, \tag{4.17}$$

one deduces that orthogonal projections of Eq. 4.15 are merely the orthogonal projection on the sphere and hence Eq. 4.13 is fulfilled (Fig. 4.4 illustrates the steps of the proof). □

4.2.2 Projection from Cylinder's Tangents to the Cylinder

In the final step we desire the projection of the updated piecewise smooth curve from the cylinder's tangents to the cylinder. Such projection is easily done through normalization of the orientation variables, namely by the following map (cf. Fig. 4.4 - right):

$$(u_1,u_2,u_3,u_4,u_5,u_6) \longmapsto \left(\frac{u_1}{\|u_1^2+u_2^2+u_3^2\|}, \frac{u_2}{\|u_1^2+u_2^2+u_3^2\|}, \frac{u_3}{\|u_1^2+u_2^2+u_3^2\|}, u_4, u_5, u_6\right).$$

4.3 Finite Singularity-free Cover

The obtained variation of the initial path is compact in $\mathbb{R}^6$ and hence has a finite open cover. This leads one to the idea of a finite singularity-free cover. Such a cover is consisted of *singularity-free balls* which are extensively discussed in Chapter 3. In this context we approach the concepts of *inclusion* and *exclusion* of breakpoints. In the following discussions, r^i stands for the distance to the closest pedal point with respect to breakpoint p^i, guaranteeing the existence of the singularity-free ball $N_{r^i}(p^i)$.

4.3.1 Inclusion of Breakpoints

For each two consecutive breakpoints, namely p^i and p^{i-1}, the algorithm checks whether singularity-free balls $N_{r^i}(p^i)$ and $N_{r^{i-1}}(p^{i-1})$ cover the segment in between. If the segment is not fully covered the algorithm implements a new breakpoint in exactly the midway of uncovered part of the segment as depicted in Fig. 4.5 - left.

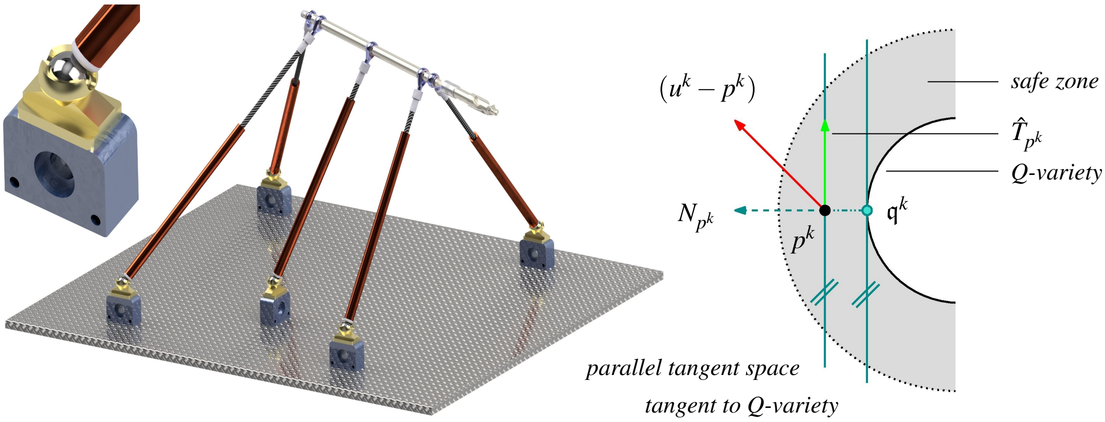


FIGURE 4.6: Left: Illustration of a simple pentapod with the LP-property in addition to a magnified view of the base spherical joints. Base joint limit is modelled by a cone with an axis orthogonal to the base plane. Right: Imaginative illustration of the projection on a tangent of a Q-variety.

4.3.2 Exclusion of Breakpoints

While the previous step assures one of singularity freeness of the curve, it is adequate not to have excessive included breakpoints. The initial idea is to exclude a breakpoint p^i if it is covered by the adjacent balls $N_{r_{i-1}}(p^{i-1})$ and $N_{r_{i+1}}(p^{i+1})$ (cf. Fig. 4.5 - right). However, one should be mindful of the fact that the breakpoint p^i itself creates the ball $N_{r_i}(p^i)$ which along with $N_{r_{i+2}}(p^{i+2})$ play the role of the cover for p^{i+1}. Deleting p^i then may cause the removal of its corresponding ball and hence a blown possible cover for p^{i+1}. In order to avoid such an undesirable situation the algorithm first labels the breakpoints which are doubly covered by adjacent balls (a point p^i is doubly covered if it belongs to $N_{r_{i-1}}(p^{i-1}) \cap N_{r_{i+1}}(p^{i+1})$, see Fig. 4.5 - right). Then the algorithm groups such breakpoints into *packs* based on the number of consequent occurrence of double covers. Calling the number of breakpoints in each pack the *size* of the pack, the algorithm takes the following strategy:

- Packs of size 1 are deleted,
- In the packs of size bigger than 1 the breakpoints with odd numbers are removed.

Finally, the algorithm repeats these steps till no pack can be found [1]. Naturally, in the above process the start- and end-pose are entirely neglected.

4.4 Joint Analysis

From the computational kinematics point of view, feasibility of optimized motion by the joint and linkages is of utmost importance. These physical limits restrict the end-effector movements and are of two types, namely, *spherical joint limits* and *prismatic joint limits*. It turns out that dealing with these restrictions can be translated into the language of variational path optimization. In fact, as it will be described in the coming subsections, it is possible to think of these restrictions as hypersurfaces in $\mathbb{R}^6$. One last remark before plunging into the details, would be regarding the design of the pentapod. In the theorems to come within this section, it is not necessary to restrict ourselves to *simple pentapods* since they are valid for *general linear pentapods* in which the base anchors belong to $\mathbb{R}^3$. Hence, the findings in this section are more general and can be considered for a more general framework as well.

[1] In practice, due to computational reasons that are discussed in Appendix C the algorithm is set to keep 6 number of breakpoints at least.

4.4.1 Prismatic Joint Limits

In the position space a point on the prismatic extendible part can be located within two spheres with radii ρ_{min} and ρ_{max}. Considering this setting in pose space $\mathbb{R}^6$ results the following theorem:

Theorem 10. *Assume a "general linear pentapod" with minimum possible prismatic length $\rho_{min} > 0$ and maximum ρ_{max}, then the set of all feasible poses with $\rho_{min} \leq \rho \leq \rho_{max}$ forms a "smooth hyperquadric" in $\mathbb{R}^6$.*

Proof. Writing down the equations of the sphere with the radius $\rho_{min} \leq \rho \leq \rho_{max}$ around a base spherical joint with coordinates $(x_i, y_i, z_i)^T$ gives

$$f_p := (r_i u_1 + u_4 - x_i)^2 + (r_i u_2 + u_5 - y_i)^2 + (r_i u_3 + u_6 - z_i)^2 - \rho^2 \tag{4.18}$$

once again by considering $f_p \in \mathbb{C}[u_1, u_2, u_3, u_4, u_5, u_6]$, Eq. 4.18 describes a quadric hypersurface in pose space $\mathbb{R}^6$. Calculating $\nabla f_p = 0$, one observes that it vanishes on $\mathbf{V}(f_p)$ iff $\rho = 0$. Hence $\mathbf{V}(f_p)$ is a smooth algebraic variety. □

4.4.2 Base Spherical Joint Limits

Theorem 11. *Assume a "planar" pentapod with base spherical joints, whose limits can be modelled as cones of revolution with axes orthogonal to the base plane, then the set of all poses resulting in the maximum freedom of movement of the i-th leg with respect to the i-th base joint limit forms a "hyperquadric" in $\mathbb{R}^6$.*

Proof. The base spherical joints located on the plane allow motions inside a cone of revolution with vertical axis (cf. Fig. 4.6 - left). The implicit equation of such a cone with *apix angle* θ and base anchor coordinates $(x_i, y_i, 0)^T$ is:

$$f_{bc} = (u_6 + r_i u_3)^2 - \cot^2\left(\frac{\theta_i}{2}\right)\left[(u_4 + r_i u_1 - x_i)^2 + (u_5 + r_i u_2 - y_i)^2\right]. \tag{4.19}$$

One may think of Eq. 4.19 as a polynomial in $\mathbb{C}[u_1, u_2, u_3, u_4, u_5, u_6]$. Hence, the i-th base cone with an apix limit of θ_i forms a hyperquadric in $\mathbb{R}^6$. Calculating $\nabla f_{bc} = 0$, one observes that it vanishes on $\mathbf{V}(f_{bc})$ iff the platform anchor point collapse with its corresponding base anchor point. Note that due to the fact that $\rho_{min} \neq 0$ such a collapse is unreachable. □

Now, the idea is to implement new terms into the cost function in such a way that if a pentapod's leg is at

- its maximum/minimum extent,
- its maximum angular limit with respect to base cones,

the algorithm allows a slide of the end-effector at the maximum/minimum extent or angular limit. Geometrically, this means that if a breakpoint p^k is located on one of the hyperquadrics (from now on called Q-variety), mentioned in Theorems 10/11, in such a way that if the update of p^k is going out of the Q-variety then the algorithm permits an update of it on $T_{p^k}(Q)$. Naturally, due to practical reasons, the algorithm must consider such breakpoint already located on Q if the breakpoint is closer than a certain positive real value ε to it (cf. Fig. 4.6 - right). Hence in such a terminology the projection on the tangent will happen only if $\langle u^k - p^k, p^k - \mathrm{q}^k \rangle < 0$, where q^k is the corresponding closest pedal to p^k on Q-variety.

Definition 34. *Assume a breakpoint p^k of a piecewise smooth path in $\mathbb{R}^6$ and a Q-variety, as mentioned in Theorems 10/11, such that $\mathfrak{d}\left(p^k, Q\right) < \varepsilon$ where $\varepsilon \in \mathbb{R}^+$. If u^k is the corresponding point to p^k after variation and if $\mathfrak{q}^k$ is the orthogonal projection of p^k on Q then define*

$$\hat{T}_{p^k} := (u^k - p^k) - \langle (u^k - p^k), N_{p^k} \rangle_g N_{p^k}, \tag{4.20}$$

where $N_{p^k} := \left(p^k - \mathfrak{q}^k\right) / \|p^k - \mathfrak{q}^k\|_g$. Then the k-th breakpoint's update should be replaced by $\hat{T}_{p^k}$.

Remark 19. *In the variational path optimization process it often happens that a breakpoint p^k breaches the ε vicinity of more than one Q-variety (e.g. by almost reaching the maximum extent of more than one leg). In such a situation the vector $(u^k - p^k)$ should be projected on the intersection of the corresponding tangent spaces. Computationally, consider the vector space V spanned by the vectors $\{(p^k - q_1^k), \ldots, (p^k - q_m^k)\}$ where $\forall i, 1 \leq i \leq m$, q_i^k are the corresponding pedal points on the Q-varieties whose ε vicinities are breached by p^k. Then the update vector $(u^k - p^k)$ should be projected on $V^\perp$. Reshaping the basis of V into the rows of a $m \times 6$ matrix A and having in mind that g is the object-oriented metric tensor matrix, one can compute the basis of $V^\perp$ using the following* MATLAB *code (the* ***null*** *command is based on SVD, cf. Section 1.2.7):*

```
U = null(A * g);
proj = (u(k,:)-p(1,:)) * U;
u(k,:) = sum(bsxfun(@times, proj, U), 2);
```

4.5 Variational Path Optimization Algorithm's Manual

This section represents a short manual of using the variational path optimization algorithm. The manual is accompanied by an algorithm's flowchart given in Appendix B.

4.5.1 Input

The algorithm will ask for two main sets of inputs namely, *design parameters* and *optimization parameters*.

Design parameters

These variables define the architectural structure of the pentapod and hence its singularity variety. These variables of **r**, **X** and **Y** stand as the defining parameters of the base and platform anchor points while the parameters α and β describe either the pattern of platform anchor points (in the LP-case) or the co-linearity of certain number of base anchor points (in LO-cases). In a more general terminology, the design parameters inform the algorithm on building up the geometric obstacle (i.e. Σ-variety).

Optimization parameters

These variables globally control the variations of the initial curve. These parameters are read as follows:

- n: the number of breakpoints required for the piecewise smooth curve,
- λ: the geodesic weight,

- η: the bending weight,
- growth: is a percentage value indicating the maximum and minimum permitted variation of the curve at each iteration. Experimentally, it is recommended to be ±5%.

4.5.2 Process I

The main task of *process I* is to create a data collection of the inputs. In fact, these are the calculations that the algorithm needs to do to have an initial understanding of the situation of the initial path and the singularity variety (obstacle). This process is done only *once* per run. During this process the manipulator will not perform any action and hence the *clock* will not *tic*.

1. Retrieving the data on the given singularity-free initial path within the joint limits.

2, 3. The algorithm evaluates the situation of the initial curve with respect to the Σ-variety (obstacle) by obtaining the corresponding pedal points on the Σ-variety (obstacle). This act is done in the fashion of the following consecutive steps:

- finding the real pedal points (cf. Section 1.1.8),
- finding distances corresponding to the real pedal points,
- sorting the real pedal points with respect to their distance in such a way that the closest pedal point is labelled as the first.

4. The algorithm calculates the minimal singularity-free cover by including/excluding breakpoints (*upon user's request*).

5. At this point, the algorithm runs a check on joint rate limits. Additionally, the safe zone breaches are flagged (*upon user's request*).

6. The objective function is evaluated. In the coming iterations one must expect the monotonic descent of the objective function per iteration till convergence at a possible local minimum.

4.5.3 Decision I

Though *Decision I* could be written as a *while loop*, due to imposing more control over the outcomes by the user (i.e. user can call a result before the final convergence) it is presented as a *for loop* (cf. Appendix B). Additionally, by the start of *Decision I*, the clock *ticks* to measure the *elapsed time*.

4.5.4 Process II

The core of the algorithm is located within the *Process II*.

1. By solving the linear system, the preferred direction $u - p$ is obtained.
2. The algorithm monitors whether the safe zone of Q-varieties is breached or not. If the answer is positive then the update of breakpoints responsible for the breach is done in the fashion of Eq. 4.20 (cf. Remark 19).

3, 4, 5. Computing the step size as explained in Section 4.1.3.

6. Having the step size, the curve's new pose is updated as follows:

```
up(2:n-1,:) = p(2:n-1,:) + (step_size) * (u(2:n-1,:));
```

7, 8. Back-projection into Γ as explained in Section 4.2.

9. As we project the result into Γ there is no guarantee for the minimal singularity-free cover to remain intact. Hence, based on the two algorithms described in Section 4.3, the inclusion and exclusion should be redone.

10. The cost function is calculated for the current iteration label $i+1$.

4.5.5 Decision II

The new cost function, $\mathsf{cost\_function}(\mathsf{i}+1)$ is compared with the previous one, namely $\mathsf{cost\_function}(\mathsf{i})$. Since by the gradient descent it is expected of the cost function per iteration to decrease, in *Decision II* a *while loop* is commenced. As long as the cost function is not descending and the value of step size is not *zero* (this can be changed by user to a small number such as 10^{-6} for being real-time preservation) the algorithm executes the *while loop*. If the cost function's value is remained intact (almost intact with respect to an accuracy of 10^{-6}) for a certain number of iteration the algorithm terminates.

4.5.6 Process III

At each iteration of the *while loop* the algorithm halves the step size and reruns *Process II*. This gives the opportunity for the preferred direction to find the proper direction to a local minimum.

4.5.7 Output

The coordinates of the *final curve*, the plot of the optimized motion and the *objective function per iteration diagram* are printed.

4.6 Results & Discussions

The results are demonstrated in the form of the following examples. The algorithm used to produce these results is a Matlab first implementation and can be subject to further improvements, especially through recoding in C++. Finally, the Matlab source codes for the algorithm are downloadable via http://www.geometrie.tuwien.ac.at/rasoulzadeh/.

4.6.1 Example I

Consider a *simple pentapod* of the 3rd-LO type (cf. Fig. 2.6, the manipulator on the left side of the cube) with the following architecture matrix (cf. Eq. 2.4):

$$\begin{pmatrix} r_2 & x_2 & y_2 & z_2 \\ r_3 & x_3 & y_3 & z_3 \\ r_4 & x_4 & y_4 & z_4 \\ r_5 & x_5 & y_5 & z_5 \end{pmatrix} = \begin{pmatrix} 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 5 & 8 & 3 & 0 \\ 9 & 12 & 12 & 0 \end{pmatrix}. \qquad (4.21)$$

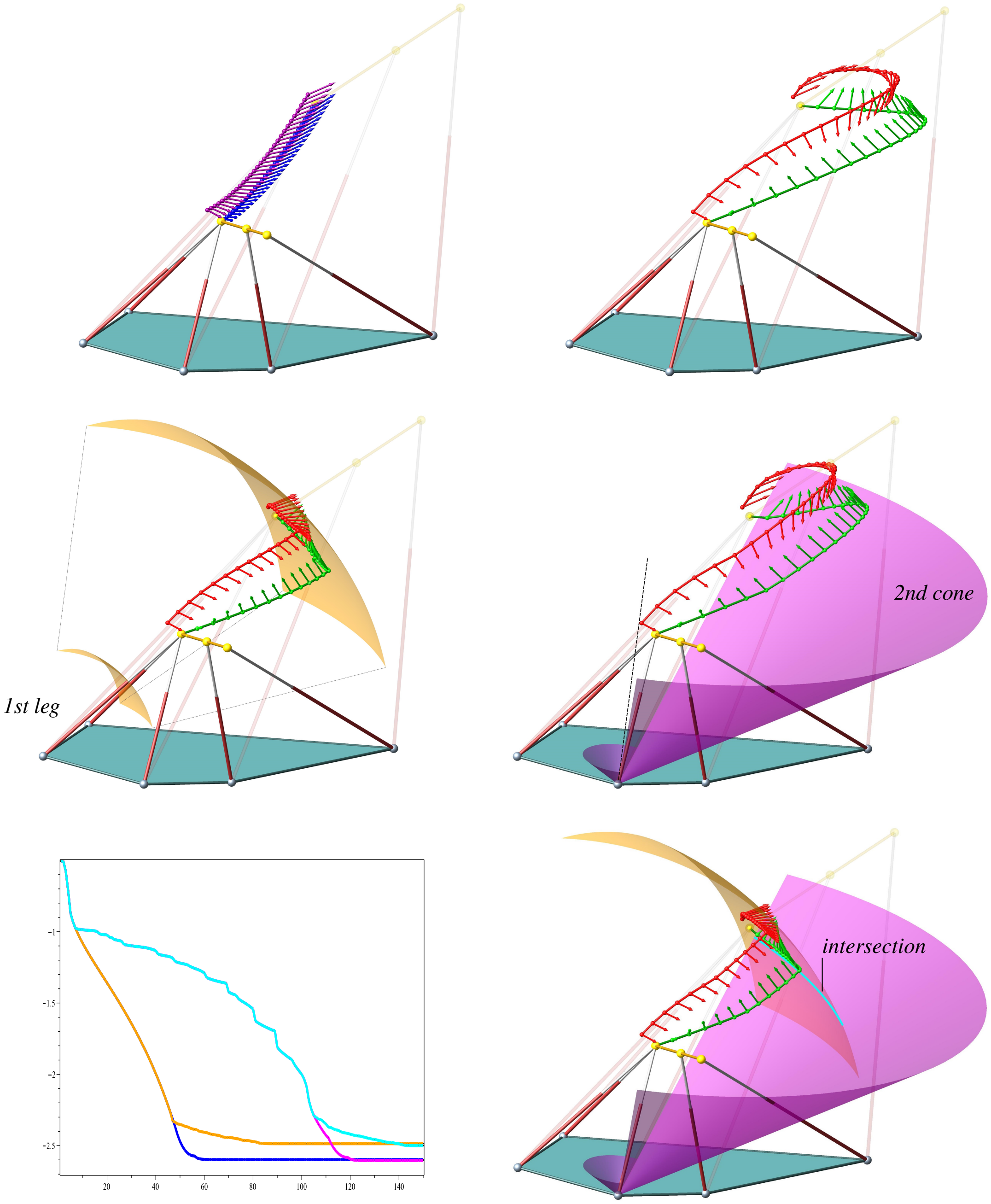


FIGURE 4.7: Illustration of the paths in $\mathbb{R}^6$ and their corresponding pedals as vector fields along paths in $\mathbb{R}^3$. Top-left: Initial path (blue) and its corresponding pedals (purple), Top-right: Final path (green) and its corresponding pedals (red) (blue objective curve), Middle-left: Modification of the final path under 1st prismatic leg's safe-zone breach (golden objective curve), two golden sphere's portions depict the minimum contraction and maximum extension of the first leg, Middle-right: Modification of final path under 2nd base cone's safe-zone breach (magenta objective curve). Bottom-right: Modification of final path under 2nd base cone's safe-zone breach and 1st prismatic leg's maximum extent safe-zone breach at the same time (cyan objective curve). Bottom-left: Objective function (vertical axis) per number of iterations (horizontal axis) descent. Note that this descent fully makes sense with respect to the geometry of the optimized motions. The animated versions of these figures are downloadable via `http://www.geometrie.tuwien.ac.at/rasoulzadeh/`.

Additionally, the second base cone's apex angle (resembling the physical limits of base spherical joints) is set at 108° while the range of the first leg's prismatic joint varies in the closed interval $[5.1, 16]$. The initial singularity-free path is given by the curve

$$\alpha : [2,5] \longrightarrow \Gamma \subset \mathbb{R}^6, \tag{4.22}$$

$$x \longmapsto \left(\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta), \frac{x+10}{3}, \frac{x^2+10}{3}, \frac{x^3}{30}+5.333\right), \tag{4.23}$$

where

$$\begin{pmatrix}\theta\\ \phi\end{pmatrix} = \frac{5-x}{3}\begin{pmatrix}0.4\,\pi\\ 6.8\,\pi\end{pmatrix} + \frac{x-2}{3}\begin{pmatrix}0.25\,\pi\\ 2\,\pi\end{pmatrix}. \tag{4.24}$$

The illustrated cases are computed for 30 breakpoints with the geodesic weight $\lambda = 0.05$ and the bending weight $\eta = 0.031$. Furthermore, the curve's growth is given by growth $= 5\%$ and safe zone's vicinity is given by $\varepsilon = 0.4$ (cf. Definition 34).
Finally, the development of the motion curve are depicted in Fig. 4.7 and the geometric details of the final optimized motion is given in Table 4.1 while Fig. 4.7-bottom left shows the descent of the related objective functions per iteration.

	Top left	Top right	Middle left	Middle right	Bottom right
Length	11.1822	18.5233	14.1764	18.2383	14.3301
Total curvature	0.2379	3.2707	2.0867	3.4071	2.3987
Elapsed time	-	1.6314 s	3.9546 s	6.7991 s	8.4724 s

TABLE 4.1: Length and total curvature are computed for the final curve in $\mathbb{R}^6$ with respect to "object oriented metric" while the elapsed time gives simply the amount of the passed time for the algorithm to find the final curve.

4.6.2 Example II

Preserving the architectural and optimization parameters mentioned in Example 4.6.1 and allowing the minimal singularity-free cover to take place, the result will be as depicted in Fig. 4.9. It is noteworthy that the final number of breakpoints will be 6. Finally, the geometric details of the motion is found in Table 4.2 and the related objective function per iteration descent in Fig. 4.9 - bottom-left.

Remark 20. *The jumps in the objective functions in Fig. 4.9 refer to the inclusion of break points. Fig. 4.8 depicts the changes of number of breakpoints for the cases under study.*

	Top left	Top right	Middle left	Middle right	Bottom right
Length	11.1822	20.6106	15.7346	16.8231	14.6214
Total curvature	0.2379	12.9031	10.5039	11.7041	7.7174
Elapsed time	-	2.3237 s	3.3512 s	3.7472 s	3.5261 s
Number of breakpoints	30	7	6	6	6

TABLE 4.2: Length and total curvature are computed for the final curve in $\mathbb{R}^6$ with respect to "object oriented metric", the elapsed time gives the amount of the passed time for the algorithm to find the final curve. The number of final breakpoints is intact as the singularity-free minimal cover was not active for this optimization.

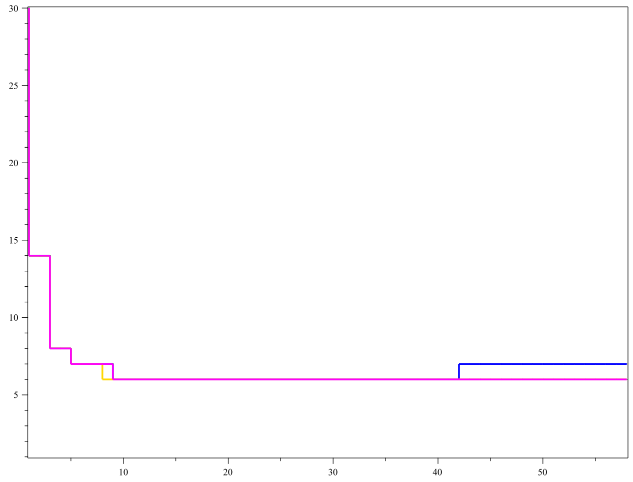


FIGURE 4.8: Illustration of changes in number of breakpoints per iteration during the exclusion/inclusion process. The blue curve depicts the case where no joint limit is applied while the golden and purple ones stand as the changes under 1st prismatic joint limit and 2nd base spherical joint limits. The cyan curve (not visible) describing the consideration of 1st prismatic and 2nd cone limit simultaneously is fully covered by the golden curve due to the similarity of change per iteration.

4.6.3 Example III

One other constructive example, to show the strength of the algorithm, is to depict how it smooths undesired non-smooth path. Assume a 3rd-LO pentapod with the same architectural design as Eq. 4.21 but with the following singularity-free given motion:

$$\alpha : [9,14] \longrightarrow \Gamma \subset \mathbb{R}^6, \tag{4.25}$$

$$x \longmapsto (\cos(T), \sin(T), 0, t-5, t-2, \arccos(\cos(t))+5), \tag{4.26}$$

where

$$T = \left(\frac{t-4}{9}\right)\frac{\pi}{2}. \tag{4.27}$$

Note that the orientation part of this motion simply sweeps a circle and the end effector is always parallel to the ground while the position curve creates a sawtooth due to the presence of $\arccos(\cos(t))$ term. Clearly such sawtooth motions are undesirable for the maintenance and safety of the manipulator and should be avoided (cf. Fig. 4.10-Top). The illustrated final optimized motion is computed for 20 breakpoints with the geodesic weight $\lambda = 0.05$, the bending weight $\eta = 1$ and a curve growth of 5% (cf. Fig. 4.10-Bottom).

	Top	Bottom
Length	11.0090	11.6501
Total curvature	1.4055	1.6386
Elapsed time	-	0.8880 s

TABLE 4.3: Length and total curvature are computed for the final curve in $\mathbb{R}^6$ with respect to “object-oriented metric", the elapsed time gives the amount of the passed time for the algorithm to find the final curve.

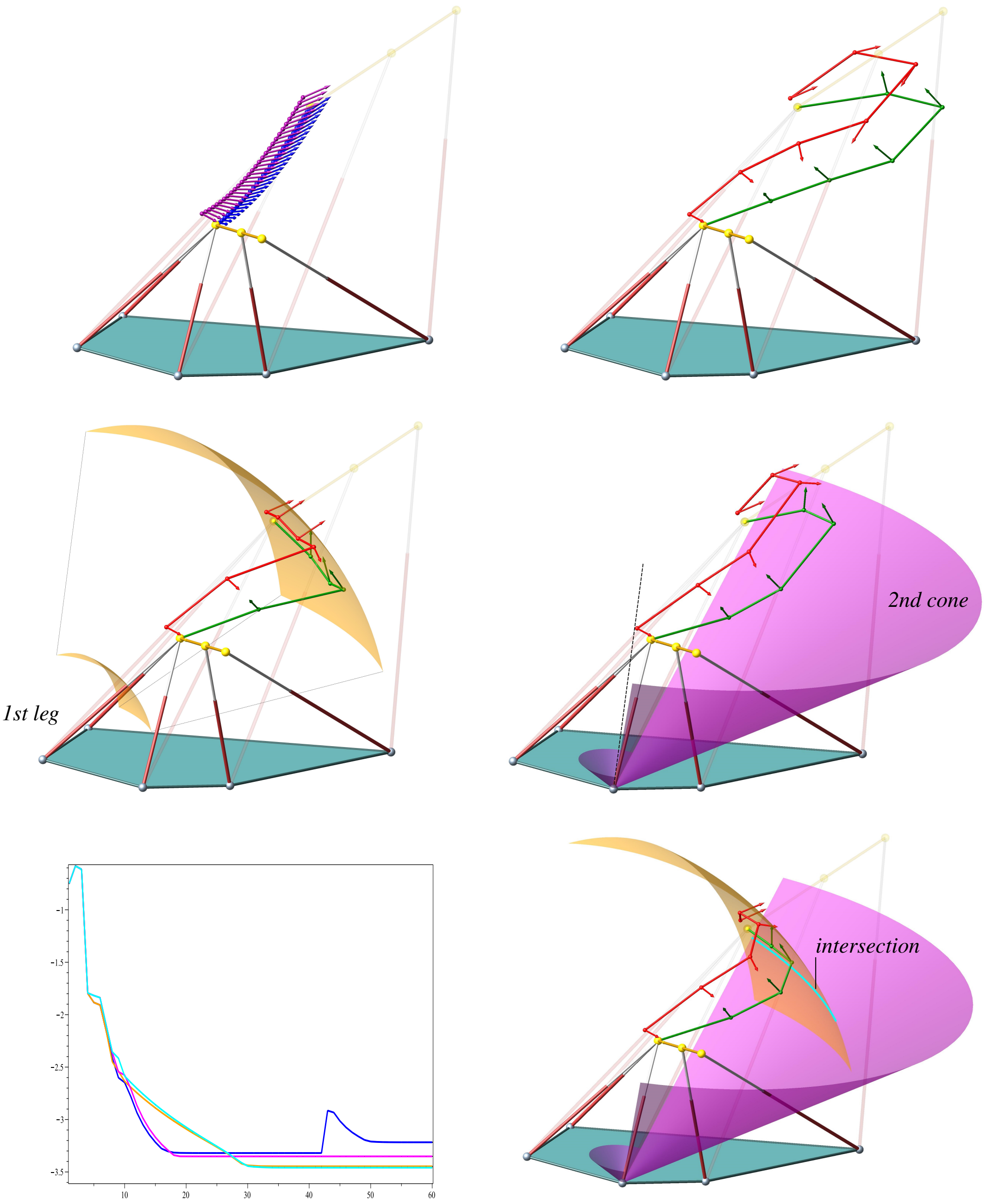


FIGURE 4.9: Illustration of the same example as in Fig. 4.7 but allowing the inclusion and exclusion of breakpoints to obtain a minimal singularity-free cover. Bottom-left: The jumps in the objective functions result from the inclusion of breakpoints. Bottom-right: The green breakpoints are not “exactly" on the intersection curve (cyan) due to the fact that a certain small gap for computational insurance has been taken into account

4.6.4 Variational Path Optimization Through Graphical User Interface (GUI)

The nature of variational path optimization algorithm is quite general, allowing it to be used for purposes other than optimization of a manipulator’s trajectory (e.g. such as optimization of path with respect to parabola or ellipse in $\mathbb{R}^2$ or higher dimensions). Henceforth, we would like to have program allowing the user to utilize this algorithm for different purposes. Widely distributed programs typically interact with their users not only through characters on keyboard and the screen but also through pictures, mouse clicks and screen touches. The

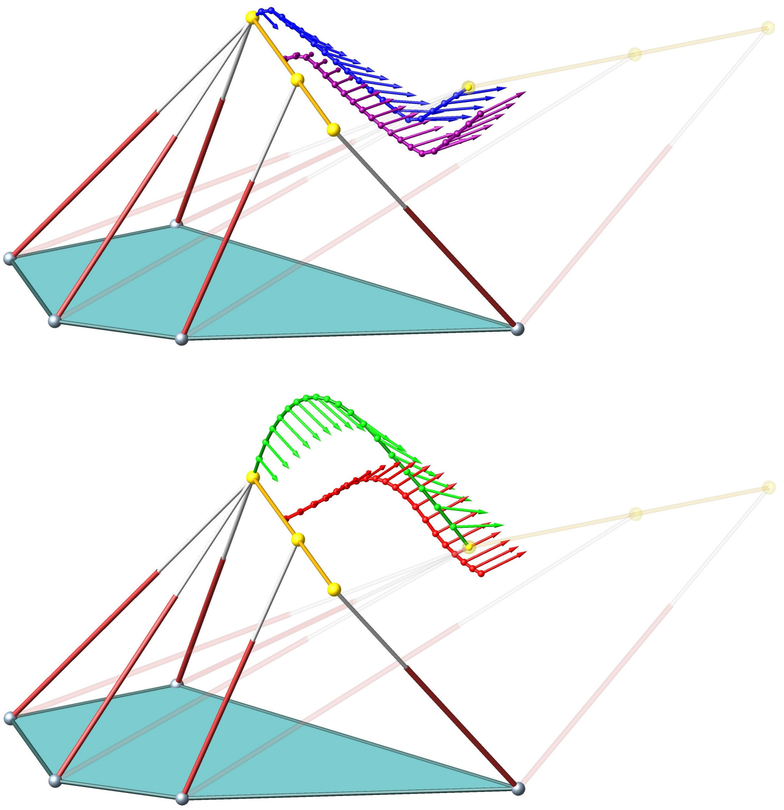


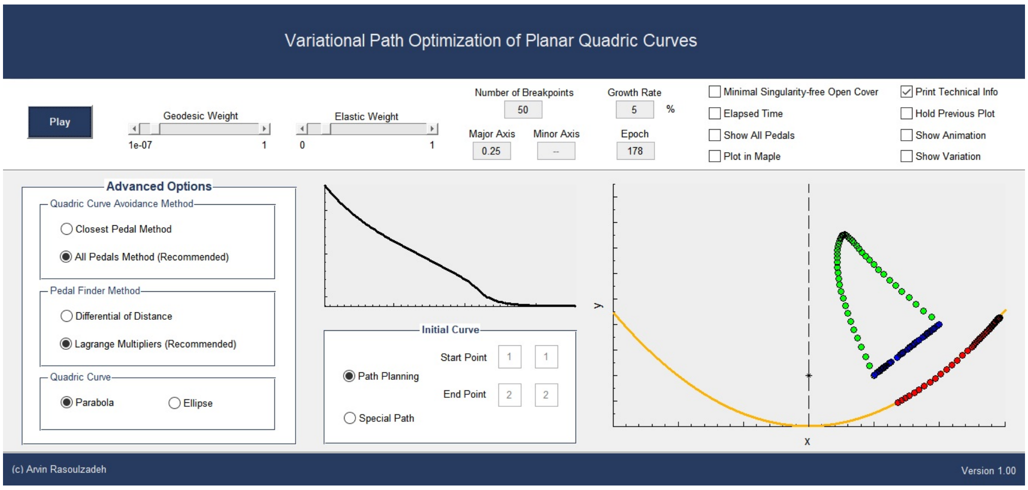
FIGURE 4.10: Illustration of the *sawtooth* motion and its smoothed optimized version. Top: Initial path (blue) and its corresponding pedals (purple), Bottom: Final path (green) and its corresponding pedals (red).

scheme of interaction that such a program presents to its users is called *graphical user interface* which is famously abbreviated as *GUI* [Fitzpatrick and Ledeczi, 2015].
As it is evident from this chapter the *variational path optimization algorithm* depends on a variety of optimization and design parameters. Typing these parameters on keyboard for a test is a time consuming task, aside from the fact that it is quite error prone. In order to alleviate such a conspicuous burden, we developed two graphical user interfaces covering the implementation of variational path optimization for two families of planar quadric curves plus the class of simple pentapods. These GUIs are created in MATLAB and the last version of their source codes are accessible via `http://www.geometrie.tuwien.ac.at/rasoulzadeh/`. Here we present a brief demonstrating documentation of these GUIs.

GUI for the Variational Path Optimization of Quadrics

The graphical user interface will appear by typing variational_UI in MATLAB command windows. The GUI contains several *control panels* (see Fig. 4.11) where their description is as follows:

1. **Initial Curve:** This panel contains four *Edit Text* regions receiving the coordinates of start/end points of the initial curve for the coming proper variation. Additionally, this panel contains a *Radio Button* which allows the user to choose an already pre-defined special path to be used for the optimization. In future updates a variety of choices in the form of pop-up menu will be made available to users.



Variational Path Optimization of Planar Quadric Curves

Geodesic Weight 1e-07 1

Elastic Weight 0 1

Number of Breakpoints 50

Growth Rate 5 %

Major Axis 0.25

Minor Axis --

Epoch 178

□ Minimal Singularity-free Open Cover

□ Elapsed Time

□ Show All Pedals

□ Plot in Maple

☑ Print Technical Info

□ Hold Previous Plot

□ Show Animation

□ Show Variation

Advanced Options

Quadric Curve Avoidance Method

○ Closest Pedal Method

● All Pedals Method (Recommended)

Pedal Finder Method

○ Differential of Distance

● Lagrange Multipliers (Recommended)

Quadric Curve

● Parabola ○ Ellipse

Initial Curve

● Path Planning

○ Special Path

Start Point 1 1

End Point 2 2

(c) Arvin Rasoulzadeh

Version 1.00

FIGURE 4.11: The graphical user interface of the variational path optimization of the quadrics.

2. **Advanced Options:** Presents a panel containing three sub-panels:

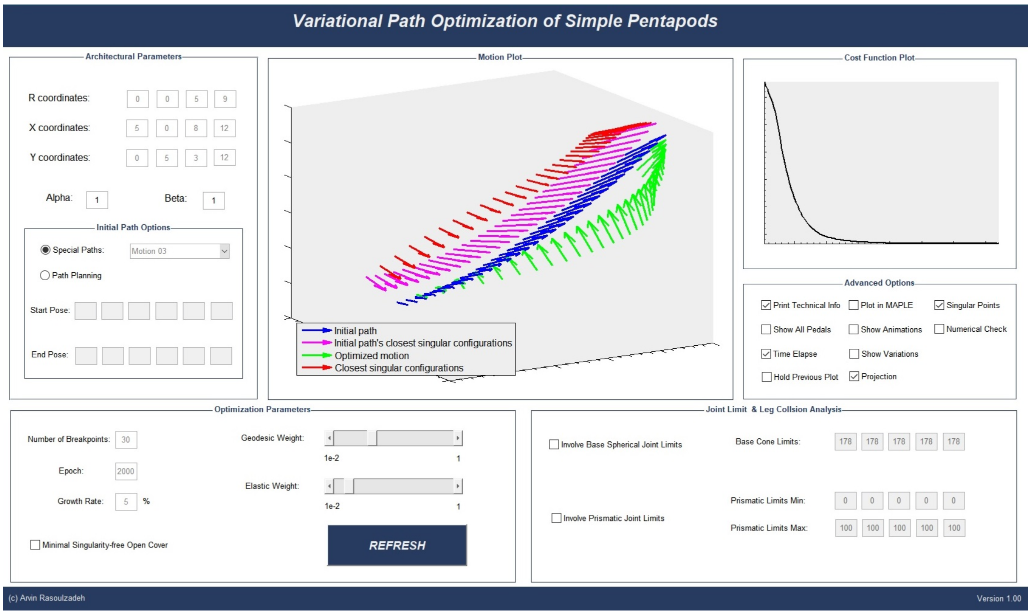
    - **Quadric Curve Avoidance Method:** Provides two options namely *Classical Pedal Method*, which considers only the closest pedal points, and *All pedals Method* which considers all pedals on the quadric in a weighted fashion,
    - **Pedal Finder Method:** Allows the user to choose between the *Differential of Distance* function and the *Lagrange Multipliers* technique to obtain the pedal points,
    - **Quadric Curve:** Presents a radio button panel which allows the user to choose between the two possible options of the ellipse and parabola. In future updates the rest of the planar quadrics will be added.

3. **The Ribbon:** Finally, after choosing the desired options for the output, on the white ribbon, on top, the user must set the optimization parameters. This part consists of two slide bars by which the user indicates the level of the desired geodesic and bending weights. There are five edit text regions which should be filled with values that indicate the following items:

    - **Number of breakpoints:** Attains any natural number between 6 and 300, indicating the chosen breakpoints on the initial curve,
    - **Growth rate:** Attains any positive real number in the interval $(0, 100]$, indicating the variational growth of updated curve. This value by default is set to 5%,
    - **Epoch:** attains any natural number between 1 and 2000, indicating the desired number of iterations. This value is by default set to 250,
    - **Curve Parameter 1:** Presents the value A in parabola $y = Ax^2$ and ellipse $x^2/A^2 + y^2/B^2 = 1$. It attains any positive real number between 0.25 and 200,
    - **Curve Parameter 2:** Presents the value B in ellipse ellipse $x^2/A^2 + y^2/B^2 = 1$. It attains any positive real number between 0.25 and 200.

    The ribbon also contains some check boxes, allowing the user to control the plottings and visibility of geometrical objects. The description is as follows:

    - **Minimal Singularity-free Open Cover:** Covers the curve at each iteration with open disks guaranteeing the lack of intersection with the quadric. The detailed explanation is given in Section 4.3,



R coordinates: 0 0 5 9

X coordinates: 5 0 8 12

Y coordinates: 0 5 3 12

Alpha: 1 Beta: 1

● Special Paths: Motion 03

○ Path Planning

Start Pose:

End Pose:

☑ Print Technical Info □ Plot in MAPLE ☑ Singular Points

□ Show All Pedals □ Show Animations □ Numerical Check

☑ Time Elapse □ Show Variations

□ Hold Previous Plot ☑ Projection

Number of Breakpoints: 30

Epoch: 2000

Growth Rate: 5 %

Geodesic Weight: 1e-2 1

Elastic Weight: 1e-2 1

□ Minimal Singularity-free Open Cover

□ Involve Base Spherical Joint Limits

Base Cone Limits: 178 178 178 178 178

□ Involve Prismatic Joint Limits

Prismatic Limits Min: 0 0 0 0 0

Prismatic Limits Max: 100 100 100 100 100

FIGURE 4.12: The graphical user interface of the variational path optimization of the simple pentapods.

- **Elapsed Time:** Prints out the run-time of the optimization process,
- **Show All Pedals:** Shows all the pedal points on the quadric,
- **Plot in Maple:** Plots the results with a high resolution in Maple.
- **Print Technical Info:** Prints the technical information of the optimization and chosen options in MATLAB command window,
- **Hold Previous Plot:** Maintains the previous plot to compare the current and the previous plot,
- **Show Animation:** Animates the point tracing the final optimized curve,
- **Show Variation:** Animates the development of the initial curve through the number of iterations till it gets its final form. This part will be available in a future release,

4. **Cost Function Plot:** This panel contains one *Axes*, allowing the user to monitor the descent of the cost function per iteration.

5. **Path Plot:** This panel contains one *Axes*, allowing the user to view the final optimized curve.

Finally, on the bottom right the *Version* of the GUI is printed. The very original version starts from 1.00 and will increase by each update.

GUI for the Variational Path Optimization of Simple Pentapods

The graphical user interface will appear by typing gui in MATLAB command windows. The GUI contains several *control panels* (see Fig. 4.12) where their description is as follows:

1. **Architectural Parameters & Initial Path Options:** This panel contains a couple of *Edit Text* regions receiving the architectural information of the manipulator. This information helps the algorithm to build the singularity variety. Additionally, this panel

contains a sub-panel called *Initial Path Options*, which itself has two *Radio Buttons* and two rows, each containing six edit texts receiving the coordinates of a start- and an end-pose of the pentapod and creates an *initial singularity-free motion*. The radio button allows one to choose between path planning option and already saved special paths to be used for further optimizations. At the moment the path planning option of the simple pentapods is disabled and will be available in a future release. For theoretical details see Section 2.4 and Section 5.1.

2. **Optimization Parameters:** This panel has three *Edit Texts*, a *Check Box*, two *Sliders* and a *push button*. The accurate description is as follows:
   - **Number of Breakpoints:** Attains any natural number between 6 and 300, indicating the chosen breakpoints on the initial curve,
   - **Growth rate:** Attains any positive real number in the interval $(0, 100]$, indicating the variational growth of updated curve. This value by default is set to 5%,
   - **Epoch:** Attains any natural number between 1 and 2000, indicating the desired number of iterations. This value is by default set to 250,
   - **Geodesic Weight:** Attains any positive real number in the interval $(0, 1]$, indicating the emphasize on geodesic energy of the motion,
   - **Elastic Weight:** Attains any positive real number in the interval $(0, 1]$, indicating the emphasize on bending energy of the motion (smoothness of the motion),

3. **Joint Limits:** This panel hosts three rows of five edit texts each, indicating the limits of prismatic joints and base spherical joints. Upon checking the corresponding boxes these physical workspace limits are taken into account during the path optimization and planning. The edit texts accept values in the range of $(0, 100]$ for the prismatic joints' extension/collapse and $[10°, 178°]$ for the apex of the base spherical joints' cone. For theoretical details see Section 4.4,

4. **Advanced Options:** Contains a series of check boxes in order to enable the user to have more control over the final plot which are demonstrated below:
   - **Print Technical Info:** Prints the technical information of the optimization and chosen options in MATLAB command window,
   - **Show All Pedals:** Shows all the pedal points on the singularity variety in the form of vector fields in $\mathbb{R}^3$,
   - **Elapsed Time:** Prints out the run-time of the optimization process,
   - **Hold Previous Plot:** maintains the previous plot for comparison,
   - **Plot in Maple:** Plots the results with a high resolution in Maple. Since Maple has several plot options, the user can convert all the optimization's data from MATLAB to Maple in order to obtain a detailed plot. Upon checking this box, after the optimization process is finished the file "plot.mw" is automatically opened. At this point the user has to open the file "pentapod.mw" manually and execute this worksheet before executing "plot.mw".
   - **Show Animation:** Animates the point tracing the final optimized curve in the form of the platform tracing the final optimized vector field in $\mathbb{R}^3$ (will be available in a future release),
   - **Show Variation:** Animates the development of the initial curve through the number of iterations in the form of development of vector fields in $\mathbb{R}^3$ (will be available in a future release),

- **Projection:** projects the optimized curve on the cylinder Γ (This box is created for designer's control over development of the GUI and is always checked. This check box is not in user's access domain.),
- **Singular points:** Considers the pedal point on the singular points of the singularity variety, namely Σ_3 (This box is created for designer's control over development of the GUI and is always checked. This check box is not in user's access domain.). For theoretical details see Section 4.1.1,
- **Numerical Check:** Numerically checks the accuracy of placement of the pedal points on the singularity variety,
- **GCA test:** Tests the singularity variety's structure through Grassmann-Cayley algebra method. (This check box is not currently in user's access domain.),

5. **Cost Function Plot:** This panel contains one *Axes*, allowing the user to monitor the descent of the cost function per iteration.

6. **Path Plot:** This panel contains one *Axes*, allowing the user to view the final optimized motion of the manipulator.

7. **Animation:** Aside from static plots, the goal is to have a full animation of the pentapod's motion in both MATLAB and Maple environment upon user's request. Fulfilling this goal allows one to have a full kinematic simulation of simple pentapods.

Once again, on the bottom right the *Version* of the GUI is printed. The very original version starts from number 1 and will be updated in future. The current manual will be updated and released as a PDF alongside the source code files under the title *Manual*.

Chapter 5

Conclusion and Future Works

In this thesis, the design, path optimization and geometry of the linear pentapod's singularity variety are thoroughly discussed and investigated. However, there are still scientifically open questions regarding the linear pentapods that could be drawn. In this last chapter these possible future projects are discussed. It is expected that using the contents of this thesis, these problems are resolvable in near future.

5.1 Analytical Path Planning of the Simple Pentapods

There are many different techniques for the path planning of manipulators in which many of them resort to numerical techniques [Li et al., 2017].
In Chapter 2, we observed that the singularity variety of a general linear pentapod is cubic in pose variables and hence is complex enough to smoke away any idea of obtaining an analytical singularity-free path. However, the case of simple pentapods may give a hope of finding such a path without the aid of numerical techniques.
In the case of simple pentapods the singularity variety breaks into the union of a hyperplane and a quadric in $\mathbb{R}^6$ (for details see Chapter 2). Since these are *simple* geometrical objects one could expect finding an analytical path planning via the concept of *road maps* [Din and Schost, 2017].
In this subsection, we present a geometrical argument which reveals the possibility of an analytical path planning of the simple pentapods.

Definition 35. *Assume $P \in \mathbb{C}[x_1,\ldots,x_n]$ then we define a "level set" of the polynomial P to be the set of all zeros of the polynomial $Q \in \mathbb{C}[x_1,\ldots,x_n]$,*

$$Q(x_1,\ldots,x_n) := P(x_1,\ldots,x_n) - C, \tag{5.1}$$

where $C \in \mathbb{R}$.

From the geometrical point of view the level set $\mathbf{V}(Q)$ has no intersection with $\mathbf{V}(P)$. This is due to the fact that

$$\langle Q, P\rangle = \langle P - C, P\rangle = \langle 1\rangle \tag{5.2}$$

which yields $\mathbf{V}(P) \cap \mathbf{V}(Q) = \emptyset$. Additionally, if the polynomial P is a quadratic polynomial then for all of the level sets with $C \neq 0$, the quadric $\mathbf{V}(Q)$ is a *smooth* variety. This can be easily proven by resorting to the projective space. In fact homogenizing Q for a non-zero C and computing its singular points reveals that these points can only exist on the (hyper)plane at infinity and hence in finite space these level sets are smooth.
Coming back to the path planning problem, we assume that we have two non-singular poses of the pentapod and we wish to connect them via a path which does not pass through the singularity variety. Since it is assumed that the start- and end-pose points are non-singular they must be located on two level sets of the simple pentapod's singularity polynomial, namely

$Q_1(u_1,\ldots,u_6)$ and $Q_2(u_1,\ldots,u_6)$. Here, the idea is to orthogonally project one of the poses into the level set of the other. However, this is not always a possibility as these poses might belong to two different *sides* of the singularity variety (e.g. that $C_1 < 0$ and $C_2 > 0$) and hence there might be no way to do it (e.g. the case when two point belong to two different sides of a plane). In order to find out when such a possibility exist we must take a closer look at the structure of the singularity variety.
The singularity variety of the simple pentapods are isomorphic to the zero sets of

$$u_6\left[u_6\left(u_1+u_2\right)-u_3\left(u_4+u_5-1\right)\right], \tag{5.3}$$

for the LO case and

$$u_3\left[u_6\left(u_1+u_2-1\right)-u_3\left(u_4+u_5\right)\right], \tag{5.4}$$

for the LP case. As mentioned in Chapter 2, the singularity variety of the simple pentapod decomposes into a hyperplane and a quadric as it can be inferred from Eq. 5.3 and Eq. 5.4. Now, assume that start-pose and end-pose belong to one side of the quadric Σ_2 ($C_1C_2 > 0$) as otherwise no singularity-free path exists. Now, orthogonally project one of the start- or end-pose into the level set of the other and let's call this level set Q. Now, it is natural to desire the normal form of this (smooth) quadric in order to have a clearer view. If H is the related *Hessian matrix* of the homogenization of the quadratic factors of Eq. 5.3 and Eq. 5.4, then resorting to *Sylvester's law of inertia for the quadric forms* [Friedberg, Insel, and Spence, 2013, page 443] gives a map

$$\phi : \mathbb{P}^6 \longrightarrow \mathbb{P}^6, \tag{5.5}$$
$$u \longmapsto E\,u, \tag{5.6}$$

where $u = (u_1 : \ldots : u_6 : u_0)$ is the homogenised pose coordinates in $\mathbb{P}^6$ and E is the matrix whose columns are obtained through orthonormalization of the set of eigenvectors of H. Under this transformation, the quadric is transformed into $H = E^T DE$, where D is a diagonal matrix containing the eigenvalues of H as the diagonal entries. By putting $x = (x_1 : \ldots : x_6 : x_0) = \phi(u)$ we obtain a normalized form isomorphic (possibly after a relabelling of the variables) to the following quadric form in the projective space $\mathbb{P}^6$

$$Q^N := -x_1^2 - x_2^2 + x_3^2 + x_4^2 + x_0^2 \tag{5.7}$$

for both LO and LP cases. Q^N forms a connected projective variety.

Remark 21. *Note that, the Sylverster's normalization of quadrics works on the smooth quadrics. Hence, we could not directly use it to obtain the normal form of the singularity variety.*

One last step is to consider the fact that after the transformation of Eq. 5.6, *the hyperplane at infinity* $\mathbf{V}(u_0)$ and the hyperplane Σ_1 ($\mathbf{V}(u_6)$ for LO case and $\mathbf{V}(u_3)$ for the LP case) now attain new coordinates. In this fashion, through the transformation of Eq. 5.6, $\mathbf{V}(u_0)$ and Σ_1 are isomorphic to

$$u_0 = -x_5 + x_6 \tag{5.8}$$
$$u_6 = -x_1 + x_2, \tag{5.9}$$

for the LO and

$$u_0 = x_1 + x_2 + x_3, \tag{5.10}$$
$$u_3 = -x_4 - x_5 - x_6 + x_0, \tag{5.11}$$

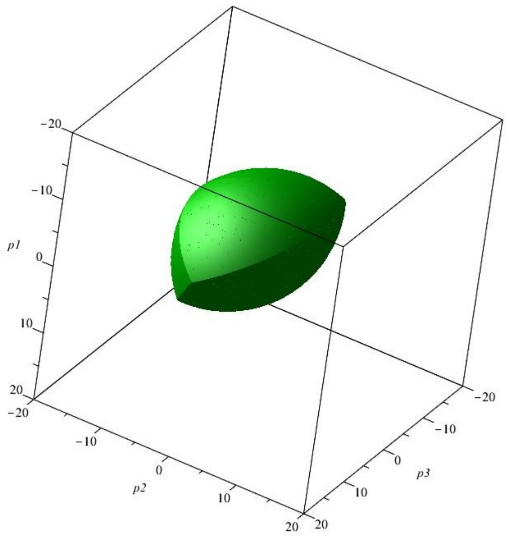


FIGURE 5.1: Illustration of the workspace in position space under constant orientation $(u_1, u_2, u_3)^T = (1,0,0)^T$.

for the LP case. Now, in order to find the path between the two end points belonging to the level set Q^N they must belong to one side of both transformed hyperplane at infinity and the transformed Σ_1 otherwise there will no possible path between them. If these conditions are fulfilled the only thing that remains is to choose the path in such a way that it does not pass through these planes. This last step requires computational considerations if it is to be *nicely* done and hence remains for a possible future project.

5.2 Workspace Analysis of Simple Pentapods

Another topic of interest related to linear pentapods is their workspace analysis. Workspace analysis takes place at its simplest whenever the five prismatic joints are confined to intervals rather than the entire positive real numbers. Since the linear pentapods are 5-DoF parallel manipulators, their workspace will be cells in $\mathbb{R}^6$ (see Section 1.2.6 for definitions of the cells). The main idea is to use an already built-in algorithm under the name of *cylindrical algebraic decompositions* to analyse these volumes.
The following example shows an initial attempt in obtaining the workspace analysis for the LO class pentapods (see Section 2.3):

Example 24. *Consider a linear pentapod possessing 1st LO property with the following architectural design:*

$$\begin{pmatrix} r_2 & x_2 & y_2 & z_2 \\ r_3 & x_3 & y_3 & z_3 \\ r_4 & x_4 & y_4 & z_4 \\ r_5 & x_5 & y_5 & z_5 \end{pmatrix} = \begin{pmatrix} 1 & 5 & 0 & 0 \\ 3 & 0 & -5 & 0 \\ 5 & -2 & -7 & 0 \\ 9 & 8 & 3 & 0 \end{pmatrix}, \tag{5.12}$$

with $\alpha = 0.2$ and $\beta = -0.2$. Now, assume that the prismatic joint's maximum length is the value 15. In another word, the each prismatic joint variable ρ_i (for $i = 1, \ldots, 5$) sweeps the interval $[0, 15]$. Now in order to be able to visualize the related volumes in $\mathbb{R}^3$ we let the manipulator attain the orientation $u_1 = 1$, $u_2 = 0$ and $u_3 = 0$.
Using the library of SIROPA, the following Maple *code gives the volume depicted in Fig. 5.1:*

```
with(SIROPA):
cell    := CellDecompositionPlus(subs(u1=1, u2= 0, u3 =0, robot:-Equations),
subs(u1=1 , u2= 0, u3 =0, [F, rho1, rho2, rho3, rho4, rho5, 15-rho1, 15-rho2,
15-rho3, 15-rho4, 15-rho5]), [rho1, rho2, rho3, rho4, rho5], [u4, u5, u6]):
```

In the above code F stands as the singularity polynomial of Eq. 2.8 and rho *stand as the prismatic joint limits. Hence, the command creates the volume that is confined by between upper bound and lower bound of the prismatic joints while the singularity polynomial is also taken into account.*

The above procedure, is repeatable for other main axis of the orientation space, namely, $(0,1,0)^T$ and $(0,0,1)^T$. However, as soon as we use a mixed orientation (which is not one of the main axis but rather is a generic point on $\mathbf{S}^2$) the computation collapses. One of the possible future projects is to investigate these volumes in a proper way both on position and orientation space.

5.3 Leg Collision Analysis and Platform Spherical Joints Limits

We demonstrated that simple pentapods imply a reduction in the number of pedal points in addition to their closed-form coordinates (cf. Section 1.1.8). We studied the geometric properties of simple pentapod's loci (cf. Section 2.4).
Since the main purpose of the paper was to optimize a given singularity-free path between two fixed poses, we setup a cost function involving energy terms and distance to the Σ-variety (cf. Section 2.4.1). Obtaining the local minimum of the cost function by imploring the gradient descent method resulted in an optimized motion with the following properties:

- increased distance to the Σ-variety,
- smoothness,

while considering the extension/contraction limits of prismatic joints and angular limits of the base spherical joints. Finally, the details of the algorithm along with a flowchart are given in Section 4.5 and Appendix B.
Though the problem of feasible motions with respect to physical limits of the base spherical joints and prismatic joints yields rather simple surfaces, namely, hyperquadrics in $\mathbb{R}^6$, the problem of *leg collision avoidance* and angular limits of *platform spherical joints* implore a more sophisticated situation as they yield hyperquartics in $\mathbb{R}^6$. Consequently, obtaining pedals on such varieties gets more complex as even the Gröbner basis approach does not yield a solution in the general case. One idea to ameliorate this problem is by resorting to a numerical method such as *homotopy continuation method* performed by Bertini [Bates et al., 2013] which will be subject to a future research.

Summarized Open problems

As the object-oriented distance $\mathfrak{d}$ is of interest for many tasks such as quality index for path planning/optimization or radius of the maximal singularity-free hypersphere the following items are still the open problems and can be considered as a generalization of the design and path planning/optimization of the simple pentapods:

- Fast computation of the pedals points (`Bertini`) alternatively we can look for the minimizer under similarity transformations of ℓ ($\Leftrightarrow$ omitting the normalizing condition $u_1^2+u_2^2+u_3^2=1$) for the class of *general linear pentapods*,
- Leg collision avoidance for the classes of *simple* and *general pentapods*,

- Motion platform angular limits consideration for the classes of *simple* and *general pentapods*,
- Workspace analysis of *simple* and *general pentapods*,
- Analytical path planning of *simple pentapods*.

Having the above cases included results in a consistent and thorough understanding of the kinematics of the linear pentapods of all classes.

Appendix A

Derivation of Pentapod's Jacobian Matrix

Kinematic singularities occur whenever the Jacobian matrix **J** becomes rank deficient, where **J** can be written as follows (cf. [Borràs, Thomas, and Torras, 2010b]):

$$\mathbf{J} = \begin{pmatrix} \mathbf{l}_1 & \dots & \mathbf{l}_5 \\ \hat{\mathbf{l}}_1 & \dots & \hat{\mathbf{l}}_5 \end{pmatrix}^T \text{ with } \mathbf{l}_j = \begin{pmatrix} u_4 + r_j u_1 - x_j \\ u_5 + r_j u_2 - y_j \\ u_6 + r_j u_3 - z_j \end{pmatrix}, \hat{\mathbf{l}}_j = \begin{pmatrix} z_j(u_5 + r_j u_2) - y_j(u_6 + r_j u_3) \\ x_j(u_6 + r_j u_3) - z_j(u_4 + r_j u_1) \\ y_j(u_4 + r_j u_1) - x_j(u_5 + r_j u_2) \end{pmatrix}.$$

As $(\mathbf{l}_j, \hat{\mathbf{l}}_j)$ are the Plücker coordinates of the jth leg, the condition $rk(\mathbf{J}) < 5$ is equivalent with the statement that the five legs belong to a linear line congruence. Now the idea is to add a sixth line in a way that it does not belong to this line congruence for all poses of ℓ. The simplest way for doing that is to consider the ideal line of a plane perpendicular to ℓ, which has Plücker coordinates:

$$(\mathbf{l}_6, \hat{\mathbf{l}}_6) := (0, 0, 0, u_1, u_2, u_3). \tag{A.1}$$

This line cannot belong to the line congruence because it does not intersect the linear platform ℓ (cf. [Pottmann and Wallner, 2009]). Therefore $rk(\mathbf{J}) < 5$ is equivalent to $\det(\mathbf{J}_+) = 0$ with

$$\mathbf{J}_+ = \begin{pmatrix} \mathbf{l}_1 & \dots & \mathbf{l}_6 \\ \hat{\mathbf{l}}_1 & \dots & \hat{\mathbf{l}}_6 \end{pmatrix}^T. \tag{A.2}$$

By embedding this matrix into a 7 × 7 matrix

$$\mathbf{S}^* = \begin{pmatrix} 1 & \mathbf{o} \\ \mathbf{o} & \mathbf{J}_+ \end{pmatrix} \tag{A.3}$$

with $\det(\mathbf{J}_+) = \det(\mathbf{S}^*)$ and by applying row and column operations to $\mathbf{S}^*$, using Maple "Insert Code Edit Region" notations, we end up with the matrix **S**:

```
with(LinearAlgebra):
x[1] := 0: y[1] := 0: z[1] := 0: r[1] := 0:
for i from 4 to 7 do
        S* := RowOperation(S*, [i, 2], -1)
end do:
for i from 4 to 7 do
        S* := RowOperation(S*, [i, 1], -r[i-2])
end do:
S* := ColumnOperation(S*, [2, 1], u1):
S* := ColumnOperation(S*, [3, 1], u2):
S* := ColumnOperation(S*, [4, 1], u3):
S* := ColumnOperation(S*, [5, 4], -u5):
S* := ColumnOperation(S*, [5, 3], u6):
S* := ColumnOperation(S*, [6, 4], u4):
```

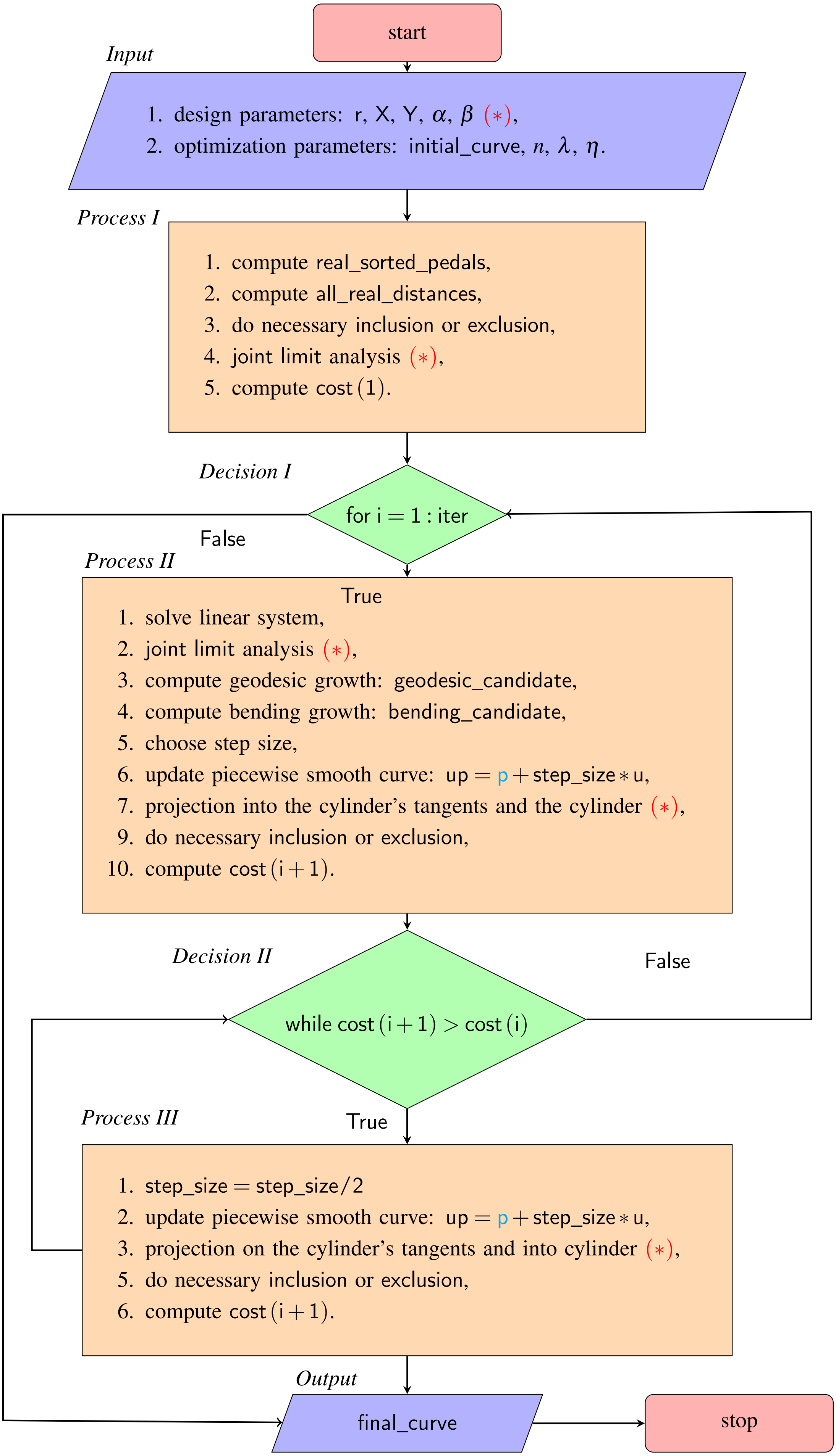
```
S* := ColumnOperation(S*, [6, 2], -u6):
S* := ColumnOperation(S*, [7, 2], u5):
S* := ColumnOperation(S*, [7, 3], -u4):
S* := ColumnOperation(S*, 6, u2):
S* := ColumnOperation(S*, [6, 5], u1):
S* := ColumnOperation(S*, [6, 7], u3):
S* := ColumnOperation(S*, 6, 1/(u1^2+u2^2+u3^2)):
S* := ColumnOperation(S*, [5, 6], -u1):
S* := ColumnOperation(S*, [7, 6], -u3):
for i from 4 to 7 do
        S* := RowOperation(S*, i, -1)
end do:
for i from 4 to 7 do
        S* := RowOperation(S*, [i, 3], r[i-2]*y[i-2])
end do:
S* := RowOperation(S*, [1, 3], u5):
S* := RowOperation(S*, 3, u2):
S* := ColumnOperation(S*, [5, 6], u3):
S* := ColumnOperation(S*, [7, 6], -u1):
S* := ColumnOperation(S*, 5, 1/u2):
S* := ColumnOperation(S*, 7, -1/u2):
S* := ColumnOperation(S*, [5, 7], inplace = true):
```

Appendix B

Variational Path Optimization Flowchart

For downloading the implementation in Matlab plus tools for plotting results in Maple visit http://www.geometrie.tuwien.ac.at/rasoulzadeh/. The implemented algorithm might be subject to minor updates in future releases.

The flowchart is modelled according to the current standard ISO 5807. Finally, due to the fact that the variational path optimization algorithm, in its presented shape here, can be used for different optimization goals involving a path and an obstacle to avoid, the flowchart is presented as general as possible. The items exclusively related to the variational path optimization of the *simple pentapods* are labelled by $(*)$. By disregarding these lines one would be able to use the same techniques for other optimization goals such as the optimization of a path with respect to a parabola in $\mathbb{R}^2$ (cf. Fig. 4.3) or similar goals in different $\mathbb{R}^n$ spaces.


start
Input
1. design parameters: r, X, Y, α, β $(*)$,
2. optimization parameters: initial_curve, n, λ, η.
Process I
1. compute real_sorted_pedals,
2. compute all_real_distances,
3. do necessary inclusion or exclusion,
4. joint limit analysis $(*)$,
5. compute cost (1).
Decision I
for $\mathsf{i} = 1 : \mathsf{iter}$
False
Process II
True
1. solve linear system,
2. joint limit analysis $(*)$,
3. compute geodesic growth: geodesic_candidate,
4. compute bending growth: bending_candidate,
5. choose step size,
6. update piecewise smooth curve: up $=$ p $+$ step_size $*$ u,
7. projection into the cylinder's tangents and the cylinder $(*)$,
9. do necessary inclusion or exclusion,
10. compute cost $(\mathsf{i}+1)$.
Decision II
False
while cost $(\mathsf{i}+1) >$ cost (i)
Process III
True
1. step_size $=$ step_size$/2$
2. update piecewise smooth curve: up $=$ p $+$ step_size $*$ u,
3. projection on the cylinder's tangents and into cylinder $(*)$,
5. do necessary inclusion or exclusion,
6. compute cost $(\mathsf{i}+1)$.
Output
final_curve
stop

Appendix C

Computational Details of the Linear System of Eq. 4.6

The linear system of Eq. 4.6 is written as $\mathbf{A}\,\mathbf{u} = \mathbf{b}$, where $\mathbf{A}$ is the *matrix of coefficients*, $\mathbf{u}$ the *matrix of variables* and $\mathbf{b}$ the *matrix of constants*. Here, in this appendix, the detailed internal structure of these matrices are investigated and the related codes are presented in the programming language of MATLAB. Throughout the rest of the appendix the variables λ and η (at i-th iteration) are defined as

```
lambda = (lambda * (n - 1) / (2 * L_prime));
eta = (eta * (n - 2) / (2 * Tau_prime));
```

where n is the number of breakpoints and L' and τ' are respectively the length and total curvature of the motion curve at the $i-1$-th iteration. Additionally, in the coming sections, J and R are the values defined in Definition 29.

C.1 Matrix of Coefficients A

Having in mind that $\mathbf{g}$ is the metric tensor (i.e. object-oriented metric tensor, cf. Eq. 3.1), n is the number of chosen breakpoints and λ and η are geodesic and elastic weights respectively, the construction of the matrix $\mathbf{A}$ is as follows:

```
A = (4 * lambda + 12 * eta) * eye(n-2);

nn = size(A,1);
for ii = 2:nn
    A(ii - 1,ii) = (-2 * lambda) + (-8 * eta);
end

for ii = 3:nn
    A(ii - 2, ii) = 2 * eta;
end

A_prime = zeros(n-2);
A_prime(1) = -2 * eta;
A_prime(end) = -2 * eta;

A = A + A_prime;
A = transpose(tril(A,2)) + A - diag(diag(A));

A = kron(g, A);
```

if d is the *dimension of the ambient space* where the *obstacle* lies then the last line of the above code is

$$\mathbf{A} = \mathbf{g}_{\,d\times d} \otimes \mathbf{A}'' \tag{C.1}$$

where $\mathbf{A}''$, is the following $d(n-2)\times d(n-2)$ square matrix:

$$\begin{pmatrix}
4\lambda+10\eta & -2\lambda-8\eta & 2\eta & 0 & & 0 & 0 & 0 & 0 \\
-2\lambda-8\eta & 4\lambda+12\eta & -2\lambda-8\eta & 2\eta & & \vdots & \vdots & \vdots & \vdots \\
2\eta & -2\lambda-8\eta & 4\lambda+12\eta & -2\lambda-8\eta & & 2\eta & 0 & 0 & 0 \\
0 & 2\eta & -2\lambda-8\eta & 4\lambda+12\eta & \ddots & -2\lambda-8\eta & 2\eta & 0 & 0 \\
0 & 0 & 2\eta & -2\lambda-8\eta & \ddots & 4\lambda+12\eta & -2\lambda-8\eta & 2\eta & 0 \\
0 & 0 & 0 & 2\eta & & -2\lambda-8\eta & 4\lambda+12\eta & -2\lambda-8\eta & 2\eta \\
\vdots & \vdots & \vdots & \vdots & & 2\eta & -2\lambda-8\eta & 4\lambda+12\eta & -2\lambda-8\eta \\
0 & 0 & 0 & 0 & & 0 & 2\eta & -2\lambda-8\eta & 4\lambda+10\eta
\end{pmatrix}$$

It is noteworthy that for number of breakpoints lower than 6, the pattern of Eq. C.1 for matrix of constants does not hold.

C.2 Matrix of Constants b

The structure of the matrix of constant is as follows:

$$\mathbf{b} = \frac{\mathbf{N}_1 \circ \mathbf{W}_1 \circ \mathbf{X}_1 + \cdots + \mathbf{N}_4 \circ \mathbf{W}_4 \circ \mathbf{X}_4}{n-2} + \mathbf{Y}, \tag{C.2}$$

where $\circ$ stands as the *Hadamard product*. The rest of the symbols in Eq. C.2 are described below.

$\mathbf{Y}$ is the $d\,(n-2)\times 1$ defined by

$$\mathbf{Y} = \left(\ (\mathbf{Y}^{1}_{n-2\times 1})^{T}, (\mathbf{Y}^{2}_{n-2\times 1})^{T}, \cdots, (\mathbf{Y}^{d}_{n-2\times 1})^{T}\ \right) \tag{C.3}$$

where each $\mathbf{Y}^{i}$ is the row of the following matrix Y:

```
Y = zeros(n - 2, 3);
for ii = 1:3
    Y(:,ii) = [2 * lambda * (J * a(ii + 3) + R * a(ii)) + ...
                   4 * eta * (J * a(ii + 3) + R * a(ii)); ...
               -2 * eta * (J * a(ii + 3) + R * a(ii)); ...
               zeros(n - 6 ,1); ...
               -2 * eta * (J * b(ii + 3) + R * b(ii)); ...
               2 * lambda * (J * b(ii + 3) + R * b(ii)) +
                  4 * eta * (J * b(ii + 3) + R * b(ii))];
end

YY = zeros(n - 2, 3);
for ii = 1:3
    YY(:,ii) = [2 * lambda * (J * a(ii) + a(ii + 3)) + ...
                    4 * eta * (J * a(ii) + a(ii + 3)); ...
                -2 * eta * (J * a(ii) + a(ii + 3)); ...
                zeros(n - 6 ,1); ...
                -2 * eta * (J * b(ii) + b(ii + 3)); ...
                2 * lambda * (J * b(ii) + b(ii + 3)) + ...
                   4 * eta * (J * b(ii) + b(ii + 3))];
end
```

```
Y = [Y, YY];
```

which after reshaping creates **Y**:

```
Y = reshape(Y, [], 1);
```

$\mathbf{N}_i$ is the $d\,(n-2)\,\times\,1$ matrix describing the normalized gradient filed on the Σ-variety. Suppose $\mathbf{p}$ is a $n-2\,\times\,d$ matrix containing the pose coordinates of the i-th breakpoint in its i-th row. Additionally, assume n_q to be the maximum number of pedal points on the Σ-variety (hence 4 based on Definition 30) then if $\mathbf{q}$ is a $n-2\,\times\,d\times n_q$ matrix consisting of rows, columns and *pages*, containing the coordinates of the pedal points referring to the i-th breakpoint in its i-th row then the structure of $\mathbf{N}_i$ is (in the code it is written as N1, N2, N3 and N4):

```
N = zeros(n-2, nq);
for kk = 1:nq
     vec = p(2:n-1, :) - q(2:n-1, :, kk);
    N(:, kk) = sqrt(sum(vec * g .* vec, 2));
end

N1 = kron(ones(d,1), (1./N(:,1)));
N2 = kron(ones(d,1), (1./N(:,2)));
...
Nnq = kron(ones(d,1), (1./N(:,nq)));
```

The Matrices $\mathbf{W}_i$ are the $d\,(n-2)\,\times\,1$ vectors containing the positive weights related to each breakpoint (cf. Definition 31). In order to be able to construct them one must first build the regulation factor (cf. Eq. 4.3).
If d_1, d_2, ..., d_{n_q} are the corresponding distances from the n_q number of pedal points to a breakpoint p then the *regulation factor* H is

$$H = \frac{1}{\frac{1}{d_1}+\ldots+\frac{1}{d_{nq}}}, \tag{C.4}$$

which the related vectorized code is given by

```
H = 1 ./ sum(cellfun(@sum,gdivide(1, ...
                       all_real_distances(2:n -1,:))),2);
```

Finally, with aid of H, the structure of $\mathbf{W}_i$s is as follows:

```
W1 = H .* cellfun(@sum, gdivide(1, ...
                     all_real_distances(2:n -1,1)));
W2 = H .* cellfun(@sum, gdivide(1, ...
                     all_real_distances(2:n -1,2)));
W3 = H .* cellfun(@sum, gdivide(1, ...
                     all_real_distances(2:n -1,3)));
...
Wnq = H .* cellfun(@sum, gdivide(1, ...
                     all_real_distances(2:n -1,nq)));

W1 = kron(ones(d,1), W1);
W2 = kron(ones(d,1), W2);
W3 = kron(ones(d,1), W3);
...
Wnq = kron(ones(d,1), Wnq);
```

X is the $d\,(n-2)\,\times\,nq$ defined by

$$\mathbf{X} = \Big(\ (\mathbf{X}^{\,1}_{\,n-2\times1})^T, (\mathbf{X}^{\,2}_{\,n-2\times1})^T, \cdots, (\mathbf{X}^{\,d}_{\,n-2\times1})^T\ \Big) \tag{C.5}$$

where each $\mathbf{X}^i$ is the row of the following matrix X:

```
X = zeros(n − 2, 3, nq);
for ii = 1:3
    for kk = 1:nq
        X(:, ii, kk) =  R ∗ (p(2:n − 1, ii) − ...
          q(2:n − 1, ii, kk)) + J ∗ (p(2:n − 1, ii + 3) − ...
            q(2:n − 1, ii + 3, kk));
    end
end

XX = zeros(n − 2, 3, nq);
for ii = 1:3
    for kk = 1:nq
        XX(:, ii, kk) =  J ∗ (p(2:n − 1, ii) − ...
          q(2:n − 1, ii  , kk)) + (p(2:n − 1, ii + 3) − ...
            q(2:n − 1, ii + 3, kk));
    end
end

X = [X, XX];
```

which after reshaping creates $\mathbf{Y}$:

```
X = reshape(X, [], 1, nq);
```

Now, having all the required ingredients, the final code for constructing $\mathbf{b}$ is:

```
b = (N1 .∗ W1 .∗ X(:,1) + ...
     N2 .∗ W2 .∗ X(:,2) + ...
     N3 .∗ W3 .∗ X(:,3) + ...
     ....
     Nnq .∗ Wnq .∗ X(:,nq)) / (n − 2)  + Y;
```

C.3 Matrix of Variables u

The command:

```
u = A \ b
```

solves the linear system of equation and yields the matrix of variables:

$$u = \Big(\; \left(u^{\,1}_{\,n-2\times 1}\right)^T, \left(u^{\,2}_{\,n-2\times 1}\right)^T, \cdots, \left(u^{\,d}_{\,n-2\times 1}\right)^T \; \Big)^T_{d(n-2)\times 1} \tag{C.6}$$

Bibliography

Angeles, Jorge (2002). *Fundamentals of robotic mechanical systems*. Vol. 2. Springer.

Bär, Gert F and Gunter Weiß (2006). "Kinematic analysis of a pentapod robot". In: *Journal for Geometry and Graphics* 10.2, pp. 173–182.

Bates, Daniel J et al. (2013). *Numerically solving polynomial systems with Bertini*. Vol. 25. SIAM.

Ben-Horin, Patricia et al. (2008). "SinguLab–a graphical user interface for the singularity analysis of parallel robots based on Grassmann-Cayley algebra". In: *Advances in Robot Kinematics: Analysis and Design*, pp. 49–58.

Boothby, William M (1986). *An introduction to differentiable manifolds and Riemannian geometry*. Vol. 120. Academic press.

Borràs, Júlia, Federico Thomas, and Carme Torras (2010a). "Singularity-invariant leg rearrangements in doubly-planar Stewart-Gough platforms". In: *Robotics: Science and Systems VI*.

— (2010b). "Singularity-invariant leg rearrangements in Stewart–Gough platforms". In: *Advances in Robot Kinematics: Motion in Man and Machine*, pp. 421–428.

— (2011a). "Architectural singularities of a class of pentapods". In: *Mechanism and Machine Theory* 46.8, pp. 1107–1120.

— (2011b). "Singularity-Invariant Families of Planar 5-SPU Platforms". In: *IEEE Transactions on Robotics* 27.5, pp. 837–848.

Borràs, Júlia et al. (2009). "A reconfigurable 5-DoF 5-SPU parallel platform". In: *Reconfigurable Mechanisms and Robots, 2009. ReMAR 2009. ASME/IFToMM International Conference on*. IEEE, pp. 617–623.

Carmo, Manfredo Perdigao do (1992). *Riemannian geometry*. Birkhäuser.

Coste, Michel and Seydou Moussa (2015). "On the rationality of the singularity locus of a Gough–Stewart platform—Biplanar case". In: *Mechanism and Machine Theory* 87, pp. 82–92.

Cox, David, John Little, and Donal O'Shea (2013). *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer.

Cox, David A, John Little, and Donal O'Shea (2006). *Using algebraic geometry*. Vol. 185. Springer Science & Business Media.

Dasgupta, Bhaskar and TS Mruthyunjaya (1998). "Force redundancy in parallel manipulators: theoretical and practical issues". In: *Mechanism and Machine Theory* 33.6, pp. 727–742.

Din, Mohab Safey El and Éric Schost (2017). "A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets". In: *Journal of the ACM (JACM)* 63.6, pp. 1–37.

Do Carmo, Manfredo P (2016). *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications.

Dudek, Ewa and Konstanty Holly (1994). "Nonlinear orthogonal projection". In: *Annales Polonici Mathematici*. Vol. 59. Instytut Matematyczny Polskiej Akademii Nauk, pp. 1–31.

Faugère, Jean-Charles (2010). "FGb: a library for computing Gröbner bases". In: *International Congress on Mathematical Software*. Springer, pp. 84–87.

Faugère, Jean-Charles and Daniel Lazard (1995). “Combinatorial classes of parallel manipulators”. In: *Mechanism and Machine Theory* 30.6, pp. 765–776.

Fitzpatrick, J Michael and Akos Ledeczi (2015). *Computer Programming with MATLAB*. J. Michael Fitzpatrick Ákos Lédeczi.

Fréchet, M Maurice (1906). “Sur quelques points du calcul fonctionnel”. In: *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 22.1, pp. 1–72.

Friedberg, Stephen H, Arnold J Insel, and Lawrence E Spence (2013). *Linear Algebra: Pearson New International Edition*. Pearson Higher Ed.

Hartshorne, Robin (2013). *Algebraic geometry*. Vol. 52. Springer Science & Business Media.

Huang, Z and XW Kong (1995). “Kinematic analysis on the spatial parallel mechanisms with redundant degree of freedom”. In: *Chin. J. Mech. Eng* 31.3, pp. 44–50.

Husty, Manfred and Clément Gosselin (2008). “On the singularity surface of planar 3-RPR parallel mechanisms”. In: *Mechanics Based Design of Structures and Machines* 36.4, pp. 411–425.

Jiang, Qimi and Clément M Gosselin (2009). “Determination of the maximal singularity-free orientation workspace for the Gough–Stewart platform”. In: *Mechanism and Machine Theory* 44.6, pp. 1281–1293.

Kim, Sungbok (1997). “Operational quality analysis of parallel manipulators with actuation redundancy”. In: *Proceedings of International Conference on Robotics and Automation*. Vol. 3. IEEE, pp. 2651–2656.

Kong, Xianwen and Clement M Gosselin (2001). “Generation and forward displacement analysis of two new classes of analytic 6-SPS parallel manipulators”. In: *Journal of Field Robotics* 18.6, pp. 295–304.

Lee, John M (2003). “Smooth manifolds”. In: *Introduction to Smooth Manifolds*. Springer, p. 103.

Lemaréchal, Claude (2012). “Cauchy and the gradient method”. In: *Doc Math Extra* 251, p. 254.

Li, Baokun et al. (2017). “Singularity property and singularity-free path planning of the Gough–Stewart parallel mechanism”. In: *International Journal of Advanced Robotic Systems* 14.6.

Li, Haidong, Clément M Gosselin, and Marc J Richard (2007). “Determination of the maximal singularity-free zones in the six-dimensional workspace of the general Gough–Stewart platform”. In: *Mechanism and Machine Theory* 42.4, pp. 497–511.

Merlet, Jean-Pierre (1989). “Singular configurations of parallel manipulators and Grassmann geometry”. In: *The International Journal of Robotics Research* 8.5, pp. 45–56.

— (1996). “Redundant parallel manipulators”. In: *Laboratory Robotics and Automation* 8.1, pp. 17–24.

— (2006). *Parallel robots*. Vol. 128. Springer Science & Business Media.

Milnor, John W (1950). “On the total curvature of knots”. In: *Annals of Mathematics*, pp. 248–257.

Nag, Anirban et al. (2018). “Identifying singularity-free spheres in the position workspace of semi-regular Stewart platform manipulators”. In: *Advances in Robot Kinematics 2016*. Springer, pp. 421–430.

Nakamura, Yoshihiko and Modjtaba Ghodoussi (1989). “Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators”. In: *IEEE Transactions on Robotics and Automation* 5.3, pp. 294–302.

Nawratil, Georg (2009). “A new approach to the classification of architecturally singular parallel manipulators”. In: *Computational Kinematics*, pp. 349–358.

— (2017). “Point-models for the set of oriented line-elements–a survey”. In: *Mechanism and Machine Theory* 111, pp. 118–134.

— (2018). “On the line-symmetry of self-motions of linear pentapods”. In: *Advances in Robot Kinematics 2016*. Springer, pp. 149–159.

Nawratil, Georg and Arvin Rasoulzadeh (2018). “Kinematically Redundant Octahedral Motion Platform for Virtual Reality Simulations”. In: *New Advances in Mechanism and Machine Science*. Springer, pp. 387–400.

Nawratil, Georg and Josef Schicho (2017). “Self-motions of pentapods with linear platform”. In: *Robotica* 35.4, pp. 832–860.

Ng, Andrew (2012). *Machine Learning*. COURSERA.

Nocedal, Jorge and Stephen Wright (2006). *Numerical optimization*. Springer Science & Business Media.

O’Neill, Barrett (1983). *Semi-Riemannian geometry with applications to relativity*. Vol. 103. Academic press, p. 20.

Petersen, Peter, Sheldon Axler, and Kenneth Ribet (2006). *Riemannian geometry*. Vol. 171. Springer.

Pottmann, Helmut and Johannes Wallner (2009). *Computational line geometry*. Springer Science & Business Media.

Rasoulzadeh, Arvin and Georg Nawratil (2018). “Rational parametrization of linear pentapod’s singularity variety and the distance to it”. In: *Computational Kinematics*. Springer, pp. 516–524.

— (2019a). “Linear pentapods with a simple singularity variety–part I: Determination and Redundant Designs”. In: *IFToMM World Congress on Mechanism and Machine Science*. Springer, pp. 689–698.

— (2019b). “Linear pentapods with a simple singularity variety–part II: Computation of singularity-free balls”. In: *IFToMM World Congress on Mechanism and Machine Science*. Springer, pp. 699–708.

— (2020). “Variational path optimization of linear pentapods with a simple singularity variety”. In: *Mechanism and Machine Theory* 153.

Rudin, Walter et al. (1964). *Principles of mathematical analysis*. Vol. 3. McGraw-hill New York.

Schicho, Josef (1998). “Rational parametrization of surfaces”. In: *Journal of Symbolic Computation* 26.1, pp. 1–29.

Selig, Jon M (2004). *Geometric fundamentals of robotics*. Springer Science & Business Media.

Shafarevich, Igor Rostislavovich and Miles Reid (1994). *Basic algebraic geometry*. Vol. 2. Springer.

Wang, Jing and Cle´ ment M Gosselin (2004). “Kinematic analysis and design of kinematically redundant parallel mechanisms”. In: *J. Mech. Des.* 126.1, pp. 109–118.

Weck, Manfred and Dirk Staimer (2002). “Parallel kinematic machine tools–current state and future potentials”. In: *CIRP Annals-Manufacturing Technology* 51.2, pp. 671–683.

White, Neil L (1994). “Grassmann-Cayley algebra and robotics”. In: *Journal of Intelligent & Robotic Systems* 11.1, pp. 91–107.

Wu, Jun et al. (2009). “Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy”. In: *Mechanism and Machine Theory* 44.4, pp. 835–849.

Zanganeh, Kourosh and Jorge Angeles (1994a). “Instantaneous kinematics and design of a novel redundant parallel manipulator”. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, pp. 3043–3048.

Zanganeh, Kourosh E and Jorge Angeles (1994b). “Mobility and position analyses of a novel redundant parallel manipulator”. In: *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, pp. 3049–3054.

Zhang, Chang-De and Shin-Min Song (1992). “Forward kinematics of a class of parallel (Stewart) platforms with closed-form solutions”. In: *Journal of Field Robotics* 9.1, pp. 93–112.