

Assistive Control for Non-Contact Machining
of Random Shaped Contours

by

Abdurrahman Eray Baran

Submitted to the Graduate School of Sabancı University
in partial fulfillment of the requirements for the degree of
Philosophy of Doctorate

Sabancı University

July, 2014

Assistive Control for Non-Contact Machining of Random
Shaped Contours

APPROVED BY:

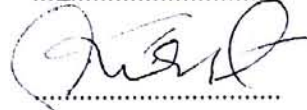
Prof. Dr. Asif Şabanoviç
(Thesis Advisor)



Prof. Dr. Ata Muğan



Assoc. Prof. Dr. Kürşat Şendur



Assoc. Prof. Dr. Ali Koşar



Assoc. Prof. Dr. Erkay Savaş



DATE OF APPROVAL: 23.07.2014

© Abdurrahman Eray Baran 2014

All Rights Reserved

Assistive Control for Non-Contact Machining of Random Shaped Contours

Abdurrahman Eray Baran

ME, Ph.D. Thesis, 2014

Thesis Advisor: Prof. Asif Şabanović

Keywords: Assistive Control, Contouring Control, Bilateral Control,
Disturbance Observer, Non-Contact Machining, Remote Laser Surgery

Abstract

Recent achievements in robotics and automation technology has opened the door towards different machining methodologies based on material removal. Considering the non force feedback nature of non-contact machining methods, careful attention on motion control design is a primary requirement for successful achievement of precise cutting both in machining and in surgery processes.

This thesis is concerned with the design of pre-processing methods and motion control techniques to provide both automated and human-assistive non-contact machining of random and complex shaped contours. In that sense, the first part of the thesis focuses on extraction of contours and generation of reference trajectories or constraints for the machining system. Based on generated trajectories, two different control schemes are utilized for high precision automated machining. In the first scheme, preview control is adopted for enhancing the tracking performance. In the second scheme, control action is generated based on direct computation of contouring error in the operational space by introducing a new coordinate frame moving with the reference contour. Further, non-contact machining is extended for realization in a master/slave telerobotic framework to enable manual remote cutting by a human operator. With the proposed approach, the human operator (i.e. a surgeon) is limited to conduct motion within a desired virtual constraint and is equipped with the ability of adjusting the cutting depth over a that contour providing advantage for laser surgery applications. The proposed framework is experimentally tested and results of the experiments prove the applicability of proposed motion control schemes and show the validity of contributions made in the context of thesis.

Rastgele Eğrilerin Temassız İşlemesi İçin Asistif Denetleyici

Abdurrahman Eray Baran

ME, Doktora Tezi, 2014

Tez Danışmanı: Prof. Dr. Asif Şabanoviç

Anahtar Kelimeler: Asistif Denetim, Çevrit İzleyen Denetim, İki Yönlü Denetim, Bozan Etmen Gözlemcisi, Temassız İşleme, Uzaktan Lazer Cerrahi

Özet

Robotik ve otomasyon teknolojilerindeki son gelişmeler, farklı işleme yöntemleri için yeni kapılar açmıştır. Temassız işleme yöntemlerinin kuvvet geri beslemesi içermeyen yapısı değerlendirildiğinde, yüksek hassasiyetli kesimin hem imalat süreçlerinde hem de cerrahi süreçlerde elde edilmesi için hareket denetim sistemi tasarımına büyük özen gösterilmesi gerekmektedir.

Bu tez, rastgele ve karmaşık şekilli eğrilerin hem otomatik hem de bilgisayar destekli manuel temassız imalatını mümkün kılacak çeşitli ön işleme yöntemlerinin ve hareket denetim tekniklerinin tasarımını ele almaktadır. Bu bağlamda, tezin ilk bölümü bu eğrilerin çıkarımına ve işleme sistemi için gerekli referans gezinmelerinin ve kısıtların oluşturulmasına odaklanmaktadır. Rastgele şekilli çevritlerin yüksek hassasiyetli otomatik işlenmesi için iki farklı denetleyici şeması üretilmiştir. Birinci şemada, özellikle keskin köşeler içeren şekillerin işleme performanslarının iyileştirilmesi adına önizleme denetleyicisi kullanılmıştır. İkinci şemada, denetleyici çıktısı görev uzayında tanımlanan ve referans çevrit ile birlikte hareket eden yeni bir koordinat sistemi üzerinden çevrit hatasının doğrudan hesaplanması temelinde oluşturulmuştur. Bir adım öteye gidilerek, önerilen yöntem temassız işlemenin, bir efendi/ köle yapısı ve telerobotik uygulama çerçevesinde bir operatör tarafından uzaktan ve manuel bir şekilde yapılmasını mümkün kılmıştır. Önerilen bu yeni yöntemle insan operatörün (örn. bir cerrahın) hareketi belirli sanal sınırlar içerisinde limitlendirilmiş ve bu sayede belirlenen çevrit üzerinde kesme derinliğinin ayarlanması mümkün kılınarak lazer cerrahi işlemler için avantaj sağlanmıştır. Önerilen çerçeve deneysel olarak gerçekleştirilmiştir ve deney sonuçları önerilen hareket denetim şemalarının uygulanabilirliğini ve tez kapsamında yapılan katkıların geçerli olduğunu kanıtlamaktadır.

Acknowledgements

I probably could have written an acknowledgement chapter to express my gratitude to some of the most brilliant people coming from all around the world who supported me throughout the long journey of my PhD study. In particular, my thesis advisor Prof. Dr. Asif Sabanovic, who, with an unbelievable experience and with an endless patience shaped my way in this research, deserves the most special thanks. His continuous encouragement and help made me overcome all the problems I faced during my doctoral study. I am greatly indebted to him and will be recalling his thoughts and advices at every instant of my future academic career.

I would also like to express my debt of gratitude to Dr. Kürşat Şendur, Dr. Ali Koşar, Dr. Erkan Savaş and Dr. Ata Muğan in accepting to be a member of my thesis jury.

It is a pleasure for me to give my very special thanks to Dr. Güllü Kızıldağ Şendur and Dr. Kürşat Şendur for their sincere help and support in shaping my academic career and for their valuable efforts in supporting my very first research experiences. Also, I am glad to express my thanks to Dr. Seta Bogosyan for creating a good environment of collaboration with her research team and to Dr. İbrahim Tekin for his valuable comments during my thesis proposal.

I am indebted to my friends Duygu Sanaç, Mine Saraç, Soner Ulun, Mehmet Ark and my colleagues Ahmet Özcan Nergiz, Ahmet Fatih Tabak, Tarik Uzunovic, Zhenishbek Zhakypov, Tarik Edip Kurt and Ahmet Kuzu for their continuous friendship, assistance and collaboration in every phase of the social and academic life during my PhD.

I would like to thank TUBITAK-Bideb and Yousef Jameel Foundation for

the financial assistance they provided me with, during the research of this thesis.

Finally, I would like to give my very special thanks to my mother Nurten, my sisters Selen and Yelin and my father Arif for making me the one I am. Especially my father, by drawing my entire way to success throughout all my educational and intellectual life, was the greatest assistance I ever had. I will be pleased to dedicate this thesis to him.

Contents

1	Introduction	1
1.1	Background & Motivation	1
1.1.1	Overview of Non-Contact Machining & Cutting	1
1.1.2	Motion Control Requirements for Non-Contact Machining Systems	4
1.2	Objectives & Goals	6
1.3	Contributions of the Thesis	7
1.4	Organization of the Thesis	10
2	Literature Review	11
3	Contour Extraction and Trajectory Generation Using an External Source	17
3.1	Image Processing Based Trajectory Acquisition	18
3.1.1	Image Enhancement and Edge Detection	19
3.1.2	Determination of Initial Data Point	21
3.1.3	Detection of Trajectory Data Points	22
3.1.4	Scaling to Operational Space	23
3.1.5	Increasing Data Resolution	25
3.2	Generation of Time Dependent Trajectories	27
3.2.1	Open Contour Time-Dependent Trajectory Generation	28
3.2.2	Closed Contour Time-Dependent Trajectory Generation	31
3.2.3	Mapping to Joint Space Coordinates	35
3.2.4	Time Based Spline Approximation	37
3.3	Generation of Time Independent Trajectories	39

3.3.1	Selection of Constraint Generating Data Set	40
3.3.2	Generation of Circular Constraints	40
4	Motion Control for Automated Non-Contact Machining	43
4.1	Background Information	44
4.1.1	Multi-DOF System Dynamics	44
4.1.2	Disturbance Observer & Robust Acceleration Control .	46
4.1.3	Mapping Between Operational and Configuration Spaces	47
4.2	Machining of Open Contour Trajectories	48
4.2.1	Preview Controller	50
4.3	Machining of Closed Contour Trajectories	51
4.3.1	Mapping Between Orthogonal and Operational Spaces	53
4.3.2	Contouring Controller	55
5	Operator Assisted Control for Non-Contact Machining	60
5.1	Constraints in Operational Space	62
5.2	Constraint Controller Derivation	63
5.3	Slave System Tracking Controller	68
5.3.1	Controller Without Time-Delay	69
5.3.2	Controller With Time-Delay	72
6	Realization	77
6.1	Background	78
6.2	Geometric Relationships	79
6.3	Position Level Kinematics	83
6.3.1	Forward Kinematics	83
6.3.2	Inverse Kinematics	87

6.4	Velocity Level Kinematics	89
7	Experimental Platform & Experiment Specific Algorithms	92
7.1	Components of Experimental Platform	92
7.1.1	Master System	93
7.1.2	Slave System	95
7.1.3	Laser Machining Unit	97
7.1.4	Image Acquisition Unit	98
7.1.5	Intermediate Components	101
7.1.6	Overall System	103
7.2	Algorithms & Methods Utilized for Experimental Realization .	103
7.2.1	Analog Position Measurement from Hall Effect Sensors	104
7.2.2	A Novel Autofocusing Method for Sharp Image Acquisition	106
7.2.3	A Novel Data Compression Method for Master Slave Communication	113
8	Experiment Results	119
8.1	Open Contour Machining Experiments	120
8.2	Closed Contour Machining Experiments	124
8.2.1	Contouring Control Experiment Set-1	124
8.2.2	Contouring Control Experiment Set-2	126
8.3	Constrained Manual Machining Experiments	128
8.4	Generalized Kinematics Validation Experiments	132
8.5	Autofocusing Experiments	135
9	Conclusion & Future Work	138

9.1	Conclusion	138
9.2	Future Work	140
10	Appendices	142
10.1	Appendix A: Realization of Constraints	142
10.2	Appendix B: Proof of invertibility of Matrix A	143

List of Figures

1.1	Illustrations of Non-Contact Machining Techniques; WJM (<i>A</i>), EDM (<i>B</i>), LM (<i>C</i>)	3
3.1	Random Shape Laser Machining Applications; Laser Engraving (<i>A</i>), Laser Microdissection (<i>B</i>)	18
3.2	Examples of Smoothed Images and Detected Edges; Smooth Open Contour Image (<i>A</i>), Smooth Closed Contour Image (<i>B</i>), Open Contour Detected Edges (<i>C</i>), Closed Contour Detected Edges (<i>D</i>)	20
3.3	Illustration of Initial Data Point Selection; User Prompted Selection for Open Contour Images (<i>A</i>), Automatic Selection for Closed Contour Images (<i>B</i>)	21
3.4	Data Point Detection; Illustration for Open Contour Image (<i>A</i>), Illustration for Closed Contour Image (<i>B</i>), Detected Data Points for Open Contour (<i>C</i>), Detected Data Points for Closed Contour (<i>D</i>)	23
3.5	Scaling from Image to Operational Space; Open Contour Data Points in Image Space (<i>A</i>), Closed Contour Data Points in Image Space (<i>B</i>), Open Contour Data Points in Task Space (<i>C</i>), Closed Contour Data Points in Task Space (<i>D</i>)	25
3.6	Increasing the Contour Resolution; Low Resolution Open Contour Data (<i>A</i>), Low Resolution Closed Contour Data (<i>B</i>), High Resolution Open Contour Data (<i>C</i>), High Resolution Closed Contour Data (<i>D</i>)	27
3.7	Schematic Illustration of Interpolation for Velocity Constrained Segmentation	30

3.8	Reference Trajectory Generation for Task-Space Coordinates. Original Image (<i>A</i>), Extracted and Enhanced Reference Trajectory Data Points (<i>B</i>), Time-Based X-Coordinate Reference Trajectory for Constant Tangential Velocity (<i>C</i>), Time-Based Y-Coordinate Reference Trajectory for Constant Tangential Velocity (<i>D</i>)	31
3.9	Representation of Closed Contour Trajectory $y(x)$ with EFDs. Number of Harmonics; 2 (<i>A</i>), 5 (<i>B</i>), 10 (<i>C</i>), 30 (<i>D</i>)	33
3.10	Time Parameterization of Closed Contour Shape. Original Task Space Trajectory and Representation with EFDs (<i>A</i>), Time-Based X-Coordinate Reference Trajectory for Constant Tangential Velocity (<i>B</i>), Time-Based Y-Coordinate Reference Trajectory for Constant Tangential Velocity (<i>C</i>)	35
3.11	Mapping from Operational Space Coordinates to Configuration Space Coordinates. X-Coordinate Reference Trajectory (<i>A</i>), Y-Coordinate Reference Trajectory (<i>B</i>), Joint-1 Reference Trajectory (<i>C</i>), Joint-2 Reference Trajectory (<i>D</i>), Joint-3 Reference Trajectory (<i>E</i>)	37
3.12	Representation of Time Independent Trajectory Generation Methodology: Illustration of Motion Constraining Circles	42
4.1	Generalized Structure of DOB for a Multi-DOF System	47
4.2	Structure of the Preview Controller	51
4.3	Schematic Illustration of Orthogonal Coordinate System Attached to the Reference Contour	52
4.4	Representation of Transformations Between Spaces	53
4.5	Structure of the Contouring Controller	59

5.1	Representative Structure of Non-Contact Assistive Machining	61
5.2	Structure of the Master Side Motion Constraining Controller .	68
5.3	Structure of the Slave System Tracking Controller Without Time Delay	72
5.4	Structure of the Slave System Tracking Controller With Time Delay	76
6.1	Schematic Representation of Delta Robot Geometry	79
6.2	Variation of angle θ_i between base and axes of prismatic ac- tuator projections: $\theta_1 = 90^\circ, \theta_2 = 190^\circ, \theta_3 = 350^\circ$ (<i>A</i>), Con- figuration with angles: $\theta_1 = 90^\circ, \theta_2 = 210^\circ, \theta_3 = 330^\circ$ (<i>B</i>), Configuration with angles: $\theta_1 = 90^\circ, \theta_2 = 230^\circ, \theta_3 = 310^\circ$ (<i>C</i>) .	80
6.3	Variation of angle α_i between base and prismatic actuators. Horizontal Configuration ($\alpha_i = 0^\circ$) (<i>A</i>), Keeps Configuration ($\alpha_i = 30^\circ$) (<i>B</i>), Vertical Configuration ($\alpha_i = 90^\circ$) (<i>C</i>)	80
7.1	The Horizontal Linear Delta Setup (Master Robot)	94
7.2	Master System; Laser Marking Pen (<i>A</i>), Master Robot with Attached Marking Pen (<i>B</i>)	95
7.3	Joint Design Used in the Slave System	96
7.4	The Keeps Linear Delta Setup (Slave Robot)	97
7.5	Illustration of Laser Machining Unit: Vertical Adjustment Sled ①, Laser Head ②, Robotic Manipulator ③	98
7.6	TIMM 400 Digital Series Microscope with Integrated Camera	100
7.7	Produced Image Acquisition Setup and its CAD Drawing . . .	101
7.8	Intermediate Components Used for Experimental Verification; Real-Time Processing Unit (<i>A</i>), Driver Electronic Boards (<i>B</i>), Supervisory Computer (<i>C</i>), Communication Medium (<i>D</i>) . . .	103

7.9	Overall Experimental System	104
7.10	Magnification Sweep with the Image Acquisition System Over a 0.05 TRY coin	107
7.11	Focusing Sweep with the Image Acquisition System Over a 0.05 TRY coin	107
7.12	Coupled Effect of Magnification and Focus over the Sharpness	108
7.13	Effect of Different Motions over Sharpness; Magnification vs. Sharpness (<i>A</i>), Focusing vs. Sharpness (<i>B</i>)	109
7.14	Maximum Focus Point Sharpness vs. Magnification Plot . . .	109
7.15	Structure of Self Optimizing Controller	113
7.16	Representative Scheme of Teleoperation System with Signal Compression	115
7.17	Representation of DCT Based Compression Scheme	116
7.18	Representation of DCT Based Decompression Scheme	117
8.1	Performance of Constant Velocity Tracking Algorithm with Preview Control. Original RGB Image of Letter "e" (<i>A</i>), Tracking Response on X-Y Plane for Letter "e" (<i>B</i>), Tangential Veloc- ity Tracking Response for Letter "e" (<i>C</i>), Tangential Velocity Tracking Error for Letter "e" (<i>D</i>)	121
8.2	Performance of Constant Velocity Tracking Algorithm with Preview Control. Original RGB Image of Letter "s" (<i>A</i>), Tracking Response on X-Y Plane for Letter "s" (<i>B</i>), Tangential Veloc- ity Tracking Response for Letter "s" (<i>C</i>), Tangential Velocity Tracking Error for Letter "s" (<i>D</i>)	122

8.3 Results from Laser Cutting Experiments for Open Contour Shapes. Original RGB Images (*A*), (*B*) (*C*), Response of Laser Cutting on Wood (*D*), (*E*), (*F*) 123

8.4 Performance of Contouring Control Algorithm. Images of CAD Generated Shapes (*A*), (*B*), (*C*); EFD Models and Tracking Responses (*D*), (*E*), (*F*); Zoomed Plots (*G*), (*H*), (*I*); Normal Direction Tracking Errors (*J*), (*K*), (*L*) 125

8.5 Performance of Contouring Control Algorithm. Images of Real Shapes (*A*), (*B*), (*C*); Detected Contours and Fitted EFD Curves (*D*), (*E*), (*F*); Laser Cutting Response on Fiberglass (*G*), (*H*); Laser Marking Response on Wood (*I*); Tangent Direction Contouring Velocities (*J*), (*K*), (*L*) 127

8.6 Original RGB Images for Constraining Master Robot Motion. Experiment-1 (*A*), Experiment-2 (*B*) 129

8.7 Constrained Remote Laser Machining Experiment 1. Master Motion (*A*), Slave Motion (*B*), Master and Re-Scaled Slave Motion (*C*), Tangential Velocity Responses (*D*), (*E*), (*F*); Fiberglass Cutting Response (*G*), (*H*); Wood Marking Response (*I*) 130

8.8 Constrained Remote Laser Machining Experiment 2. Master Motion (*A*), Slave Motion (*B*), Master and Re-Scaled Slave Motion (*C*), Tangential Velocity Responses (*D*), (*E*), (*F*); Fiberglass Cutting Response (*G*), (*H*); Wood Marking Response (*I*) 131

8.9 Experiment Platform For Verification of Generalized Kinematics 133

8.10	Generalized Kinematics Algorithm Experiment Responses. Verification of Forward Kinematics (<i>A</i>), (<i>B</i>), (<i>C</i>); Verification of Inverse Kinematics (<i>D</i>), (<i>E</i>), (<i>F</i>); Verification of Kinematic Jacobian (<i>G</i>), (<i>H</i>), (<i>I</i>);	134
8.11	Autofocusing Experiment-1. Change in Magnification Level (<i>A</i>), Change in Sharpness Level (<i>B</i>)	136
8.12	Autofocusing Experiment-2. Change in Magnification Level (<i>A</i>), Change in Sharpness Level (<i>B</i>), Process of Manual Defocusing and Automatic Focusing (<i>C</i>)	136

Nomenclature

α_*	Angle between base plane and joint axis *
β_x	Operational Space x-Axis Offset Value
β_y	Operational Space y-Axis Offset Value
β	Scaling of motion between master and slave systems
γ	Contouring controller error convergence matrix
λ	Preview controller error convergence matrix
Ψ_i	Master slave tracking controller feedback gain
ε	Error
φ	Matrix of Constraint Generating Point Set
$\mathcal{F}(\cdot)$	Forward kinematics operator
$\mathcal{I}(\cdot)$	Inverse kinematics operator
\mathcal{O}	Orthogonal direction coordinate vector
$\phi(\mathbf{x})$	Vector of operational space constraints
Ψ_*	Reference frame *
$\mathbf{A}(\mathbf{q})$	Inertia matrix
\mathbf{A}_n	Nominal inertia matrix
$\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}})$	Vector of viscous friction forces

\mathbf{C}_1	Weight of constraint error for constraint controller
\mathbf{C}_2	Weight of constraint rate error for constraint controller
$\mathbf{G}(\mathbf{q})$	Vector of gravity forces
\mathbf{H}	Matrix of low pass filters
\mathbf{J}_O	Transformation matrix between operational and orthogonal coordinates
\mathbf{J}_ϕ	Constraint jacobian
\mathbf{J}_x	Kinematic jacobian
\mathbf{K}	Constraint controller error convergence matrix
\mathbf{K}_{p1}	Preview controller position feedback gain
\mathbf{K}_{p2}	Preview controller velocity feedback gain
\mathbf{K}_{ms}	Master slave tracking controller error convergence matrix
\mathbf{K}_n	Matrix of nominal torques
\mathbf{K}_p	Contouring controller position feedback gain
\mathbf{K}_{q1}	Constraint controller position feedback gain
\mathbf{K}_{q2}	Constraint controller velocity feedback gain
\mathbf{K}_v	Contouring controller velocity feedback gain
\mathbf{P}_e	Vector of end effector coordinates
\mathbf{P}_i	Vector of coordinates for i^{th} point on the constraint trajectory

\mathbf{q}	Generalized coordinate of configuration space
$\mathbf{S}_{\mathcal{O}1}$	Weight of position error for contouring controller
$\mathbf{S}_{\mathcal{O}2}$	Weight of velocity error for contouring controller
$\mathbf{S}_{\mathcal{P}1}$	Weight of position error for preview controller
$\mathbf{S}_{\mathcal{P}2}$	Weight of velocity error for preview controller
\mathbf{T}	Generalized torque
\mathbf{u}_N	Orthogonal coordinate normal direction unit vector
\mathbf{u}_T	Orthogonal coordinate tangent direction unit vector
\mathbf{W}_1	Weight of position error for master slave tracking controller
\mathbf{W}_2	Weight of velocity error for master slave tracking controller
\mathbf{x}	Generalized coordinate of operational space
θ	EFD polar coordinate angle
θ_*	Angle between x axis of base triangle and projection of link *
a_k	k^{th} harmonic coefficient of $\cos(\theta)$ for $x(\theta)$
b_k	k^{th} harmonic coefficient of $\sin(\theta)$ for $x(\theta)$
c_k	k^{th} harmonic coefficient of $\cos(\theta)$ for $y(\theta)$
c_x	x coordinate of the center of motion constraining circle
c_y	y coordinate of the center of motion constraining circle

d_*	Distance traveled by actuator *
D_{ij}	Distance between points i and j
d_k	k^{th} harmonic coefficient of $\sin(\theta)$ for $y(\theta)$
g_i	i^{th} low pass filter gain
I_H	Height of Image Space
I_t	Time derivative of image sequence
I_W	Width of Image Space
I_x	Derivative of image in x-direction
I_y	Derivative of image in y-direction
k	Number of iterations for data point enhancement
l_*	Length of link between actuator * and corresponding edge of end effector triangle
N	Number of detected data points on the original image
O_H	Desired Height of Operational Space
O_W	Desired Width of Operational Space
R	Distance to be moved in one sample time
r	Radius of motion constraining circle
$R_{*,\dagger}$	Rotation matrix around axis * by angle †
r_b	Radius of base triangle circumcircle

r_e	Radius of end effector triangle circumcircle
T_*	Translation vector *
T_s	Sampling period
V_T	Reference tangential velocity for machining
V_x	Optical flow x-axis velocity
V_y	Optical flow y-axis velocity
x_F	x-Axis Coordinate of First Detected Point in Image Space
x_I	Image Space x-Axis Coordinates
x_i	x coordinate of i^{th} point on the constraint trajectory
x_O	Operational Space x-Axis Coordinate
y_F	y-Axis Coordinate of First Detected Point in Image Space
y_I	Image Space y-Axis Coordinates
y_i	y coordinate of i^{th} point on the constraint trajectory
y_O	Operational Space y-Axis Coordinate

Chapter I

1 Introduction

1.1 Background & Motivation

1.1.1 Overview of Non-Contact Machining & Cutting

Recent achievements in robotics and automation technology has opened the door towards different machining methodologies based on material removal. With current state of the art, manufacturing processes can be broadly divided into two groups usually referred as conventional (traditional) and unconventional (non-traditional) machining methods. Conventional machining refers to shaping a substrate material via contact forces using a harder tool and is historically an earlier stage of manufacturing that is still valid for many industries. Among the examples of conventional machining techniques, turning, boring, milling, slotting, drilling, forging and grinding can be listed.

Non-traditional machining, on the other hand, refers to non-contact material removal process using certain physical interaction phenomena between an energy source and the substrate to be machined. The energy from the source is densely localized in a very small region and directed towards the substrate. Considering the sources of energy and physical interaction types, examples

of recently developed non-contact machining techniques can be listed as waterjet machining (WJM), electric-discharge machining (EDM) and laser machining (LM). The details related to material removal mechanisms of these techniques are summarized below;

- WJM: Waterjet machining [1], [2] relies on removal of material from an incident surface using highly pressurized water sent from a nozzle with very small exit diameter. The most important advantage of WJM is the lack of thermal deformation and pollution during process. On the other hand, water contamination and limitations on nozzle exit diameter act as disadvantages preventing use of WJM for micro-sized cutting applications [3].
- EDM: In electric-discharge machining, material removal is carried out through high frequency electrical sparks over a conductor material [4]. As a commonly adopted non-contact machining method, EDM shows effective usage especially on brittle metals. The drawback of EDM is on its limited applicability only over conductive materials [5].
- LM: Laser machining is the most commonly adopted non-contact cutting methodology that has been studied for various implementations including industrial machining [6], [7], [8], product marking [9] and robotic surgery [10]. The underlying principle of laser machining is based on breaking the chemical bonds on incident surface via condensed energy of photons. A major advantage of LM is that; based on the wavelength of incident beam, a wide range of materials can be machined. Furthermore, achievement of very small spot size (i.e. in the order of microns) with precise optical instruments bring the advantage

of machining very small pieces (i.e. micro-machining) [7]. One major drawback of LM is the conic shape of laser beam that yields precision loss after a certain cutting depth.

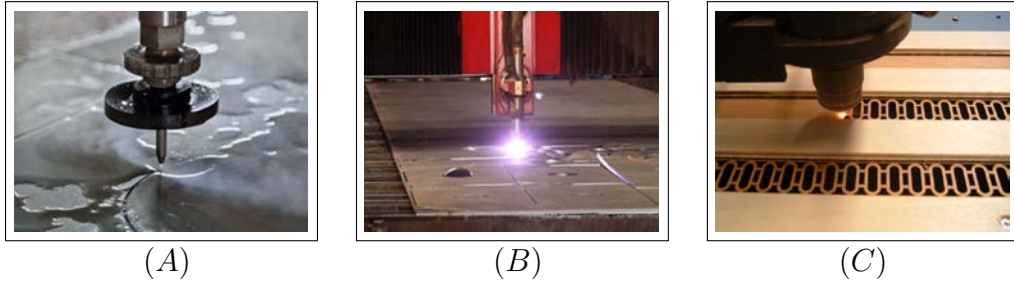


Figure 1.1: Illustrations of Non-Contact Machining Techniques; WJM (A), EDM (B), LM (C)

Examples of WJM, EDM and LM are illustrated in Figure 1.1 (A), (B) and (C) respectively. Several advantages of non-contact machining make it preferable over the conventional machining techniques. These advantages can be listed as follows;

- High accuracy and surface finishing: Due to the lack of contact force between machining tool and the workpiece, in non-contact machining, product accuracy can be enhanced to the limits of motion control system.
- Miniaturization of products: Absence of contact force also serves for shrinking the product size to the limits of control system used in machining. This opens the door for production of very small scaled (i.e. sub-micron sized) objects.
- No tool wearing: Unlike the traditional methods, in non-contact machining, there is no need for expensive hardened tools that are vul-

nerable to wearing during operation. Hence, the operational costs are further reduced with unconventional machining techniques.

- **Silent and clean operation:** The noise and dust created in traditional machining does not exist in non-contact machining. This serves for production of sensitive workpieces with the best possible accuracy.

1.1.2 Motion Control Requirements for Non-Contact Machining Systems

During non-contact machining, a nonlinear geometric path is desired to be followed by a multi-axis robotic manipulator as fast and accurate as possible [11]. Considering the motion control aspects of machining processes, satisfaction of two major objectives comes into picture. These objectives are briefly following a pre-specified trajectory as closely as possible, and maintaining a pre-specified processing speed [12]. Simultaneous enforcement of these two objectives create the main challenge in controller design of non-contact machining systems. In that sense, high performance motion control algorithms are required in modern machining systems [12].

In manufacturing processes, one of the most important issues is the reduction of machining errors to ensure the quality of final products [13]. To achieve this goal, certain desired specs exist for the motion control system. Considering the objectives mentioned above, the requirements of motion control system utilized for machining can be listed as follows;

- **High Precision:** The dimensional accuracy of the final workpiece is one important measure of product quality [14]. In that sense, accurate trajectory following capability is the fundamental requirement for motion

control system [15]. More advanced control algorithms with better resolution equipment (i.e. encoders and drivers) open the path towards improving the machining precision.

- **Rapid Operation Capability:** At low operation speeds or for low requirements on product accuracy, the tracking errors are generally acceptable. However, as the speed of production increases dynamic positioning errors start to become more effective [16]. The high speed operation of the system should be paid special attention using both low inertia actuators and advanced controllers for rapid production tasks.
- **Robustness:** The machining equipment is prone to disturbances from the surrounding environment. Due to disturbances, such as nonlinear friction or inertial loads, contour accuracy is usually deteriorated [17]. In order to preserve certain quality of production, the control algorithm should be designed in a robust way that can eliminate these disturbances during motion. Further, rigid (i.e. with less vibration) mechanical design of machining unit should also be taken into consideration for robust operation.
- **Repeatability:** The controller should be capable of maintaining certain level of processing quality regardless of the shape of final product [18]. Hence, the controller should preserve its stable and robust operation for any desired product geometry.

1.2 Objectives & Goals

The goal of this thesis is to provide a user friendly and low cost framework containing several algorithms and controllers to provide both automatic and human assisted non-contact machining of contours that have arbitrary shapes. In light of this goal, some methods and control schemes that existed in the literature have to be modified and used based on operational requirements and some new methods are required to be proposed to complete the overall picture of the framework for high precision non-contact machining. Originating from this goal, the following objectives are detailed as the target achievement of the thesis.

- *Acquisition of Motion Reference*: The first objective of this thesis is to propose a generalized and wide-range usable methodology to obtain the motion reference that can be realized in any desired system. In other words, as the primary component of a non-contact machining system, the first objective is to come up with an algorithmic structure for acquisition of desired contour information for processing. In particular, providing the intermediate steps for preparation of time-dependent motion trajectories and time-independent motion constraints regardless of the shape being traced fulfills the first objective of the thesis.
- *Automated Robust and High Precision Tracking of Desired Contour*: Once the contour reference is obtained, the second target is to come up with certain control schemes that enable high precision and robust tracking of trajectory in hand. Considering the motion control requirements of a non-contact machining system listed above, here the incorporated controllers have to preserve certain level of accuracy for

machining with the highest possible speed. In that sense, requirement is to formulate the controllers in such a way that the payoff between processing speed and machining precision is minimized.

- *Extension of the Framework for Operator Assisted Manual Machining:* Modification of controllers to include a human operator in the loop extends the application range of proposed motion control framework for non-contact machining to fields including medical applications. In that sense, another objective in this thesis is to provide certain controller schemes for high precision manual cutting of random contours by a human operator targeting to be used in laser surgery systems. Here, the controller will not only be responsible for confining the operator motion to an apriori set trajectory, but also be responsible for transferring the random motion of operator to a remote location as desired in laser surgery systems.
- *Providing Means of Practical Realization:* The final objective of this thesis is to provide the means of practical realization for the proposed trajectory acquisition and controller schemes in the context of objectives given above. In that sense, the intermediate algorithms, models and necessary derivations has to be provided in order to make the framework consistent and applicable for a variety of applications ranging from industrial production to medical operations.

1.3 Contributions of the Thesis

In the context of the study conducted in this thesis; a generalized, cheap and easy to use framework for non-contact machining of arbitrary geometries is

proposed. In that sense;

- Image processing algorithms for single image based generation of open and closed contour reference trajectories are utilized and tested in experiments. The methods cover acquisition of operational space trajectory reference from an input image and modification of the corresponding configuration space trajectory based on desired tangential velocity constraint. This way, random shaped operational space contours that do not have prior mathematical representation can be processed under a non-contact machining unit with certain desired depth.
- A control methodology is designed for robust tracking of generated trajectories based on two different schemes respectively for open and closed contour shapes. In the scheme used for open contour references, implementation of previously proposed preview control strategy is made. For closed contour references, modification of a contour tracking controller is made to act directly on contour error in operational space and provide further tracking precision. The controller is made feasible benefiting from the possibility of smooth third order differentiation using Elliptical Fourier Descriptor based mathematical representation of the corresponding closed contour reference.
- Extension of the proposed method is made for realization of remote non-contact manual machining of arbitrary shapes based on constrained control framework under a master-slave setting. In that sense, the acquired reference contained in the input image is given as motion constraint on the master system and the slave system is enforced to track the motion imposed over the master system by the human operator.

Slave system tracking is enabled for applications including communication without and with time delays using a previously proposed network observer structure.

- Considering the asymmetric mechanical structures of the produced delta robots in the experimental setup, a generalized kinematic formulation for prismatically actuated delta robots is made again in the context of this study. The proposed formulation includes a full parametric derivation of position and velocity level kinematics based on direct geometric solution which does not require iterative loops. With the proposed formulation, a whole range of kinematic configurations for linear delta manipulators is covered including the ones with asymmetric geometries, outperforming the existing "setup specific" formulations in the literature.
- For realization purposes over the experimental platform, a self optimizing controller structure is formulated to acquire the sharpest possible image from the sample under the visual inspection unit. The proposed self optimizing controller scheme is based on modification of an existing method with a continuous transition function. Further, again for practical realization purposes, an existing codec scheme is modified for application on haptic data compression for time delayed assistive machining applications. The utilized method incorporates the Discrete Cosine Transform based compression scheme with a selection algorithm to transfer the maximum power of the haptic signal to the remote system.

1.4 Organization of the Thesis

The organization of thesis is as follows. In Chapter 2, a summary of results existing in the literature are briefly discussed. In Chapter 3 image processing algorithms and methodology of trajectory and constraint generation is explained for the acquisition of random contour motion reference. In Chapter 4 controller system background is explained and two different controller structures are derived for processing of open and closed contour trajectories respectively. In Chapter 5 extension of laser machining for remote operation under the guidance of a human operator is analyzed. In that sense, motion constraining controller derivation is given along with the derivation of slave system controller for tracking of master references. For tracking of given motion references by lightweight and fast mechanisms, particular selection of prismatically actuated delta robots is made for the master and slave systems. Originating from that point, Chapter 6 analyzes and generalizes kinematic formulation for prismatically actuated delta robots providing the application flexibility over a variety of systems. In Chapter 7, the experimental platform and experiment specific algorithms used in this study is explained and in Chapter 8 results obtained from various experiments are compiled along with necessary discussion. Finally Chapter 9 gives the concluding remarks and potential opening from this research as a future study.

Chapter II

2 Literature Review

In the past few decades, non-contact machining has become an important manufacturing technique for many industrial, medical and research applications [19], [20]. Among these, the popularity of laser machining technique is growing with the increasing need for fast and precise non contact machining [21]. Unique characteristic of the laser machining is the possibility of etching or ablating exceptionally small features in many different materials with minimal damage done to the non-irradiated regions of the material [22]. Traditionally micro-scale device systems were fabricated by conventional IC or MEMS fabrication technologies [23], however, laser machining is beginning to attract more attention due to its simplicity in process, high flexibility, and high resolution [24–26].

Researchers and authors have dedicated much attention to the laser machining process modeling, simulation and applications to variety of materials [27–34]. On the other hand, system design and motion control problems specific to the laser machining systems still stay under investigation. Generally, motion control requirements for laser machining systems are high precision positioning capability [35] with nanometer resolution and repeatability along with speeds high enough to permit machining process in sufficiently

short duration [36]. Detailed reviews about the aspects of laser beam machining can be found in [37] and [38].

In certain laser machining applications, fabrication of micro channels being the most popular, constant processing depth is necessary [39]. In general, groove or channel is created by the series of laser pulses fired along the predefined trajectory. One of the ways to achieve constant depth processing with direct-write lasers is by having constant tangential positioning velocity and constant laser power. This topic has been intensively investigated by researchers and authors [40]. Generally the research has taken two main paths on the investigation of methods to reduce the contouring error. First path is concerned with the reduction of tracking errors in a single axis which indirectly reduces the contouring errors [41]. The second path taken by the researchers is concerned with the estimation of the contouring errors and contouring controller design based upon them [42].

Contour error is usually defined as the shortest distance between current position and desired path. It can be explicitly calculated for linear or circular two-dimensional contours. In conventional control systems, contour error is minimized by improving tracking accuracy of each individual axis of multi-axis system. Different control approaches can be found in [18], [43], [44] and [45]. It is known that small contour error can be achieved even when significant tracking error exists. Therefore this approach is usually used in applications where high speed tracking is not required.

On the other hand, for complex trajectories in space, contour error calculation can become a very complicated problem. Hence, free form contours can be approximated by linear segments as presented in [46] and [47]. Another approach assumes that contour is approximated by circular segments

in two-dimensional plane [42], [48], or three-dimensional space [49]. In [50], authors show how precise contour error calculation can be made if analytical description of two dimensional reference trajectory is known. An interesting concept of equivalent errors is presented in [51] for contour error expression based on the assumption that analytical description of the reference trajectory is known.

Looking from the control point of view, one of the first control strategies oriented toward reduction of contouring error is the so called cross-coupled controller (CCC), introduced more than three decades ago in [52]. The basic novelty in this approach is the design of controller that takes care of couplings between different axes during tracking of their decoupled references. Many different versions of this controller appeared in later studies. They included variable gain CCC [53], [46], [54], CCC combined with neural networks [55], self-tuning [56] and optimal control methods [57], [46]. Cross-coupled control was further modified for repetitive motion control applications, where controller learns from previous iterations in order to improve its performance [58], [59]. Introducing feed rate in the control loop brought further improvement over the performance of CCC [60], [61]. Yet, another approach in contour tracking is adopted via making use of direct contour error. In this approach authors use contour error directly to design control algorithm [51]. They do not use linear or circular segments for contour error approximation.

Recently, interesting control methods have been proposed based on coordinate transformation. In this group there exists several approaches. First one is proposed in [12], where tracking error is decomposed into normal and tangential components. Separate control actions are then applied to dynamically

decoupled system for independent tracking and feed rate control. In [16], contour tracking problem is formulated in a task coordinate frame attached to desired contour. In order to apply this coordinate transformation it is required to know feed rate, velocity direction and instantaneous curvature of the desired contour. With this approach, system dynamics is transformed to this new coordinate frame and control system assigns different dynamics to normal and tangential directions relative to the desired contour. Similarly, polar coordinate transformation is proposed in [62] where contour error is approximated with radial error. Using this simple model, contour error is taken as a state variable and contouring controller is designed by stabilizing contour error dynamics. Recently in [63] an orthogonal global task coordinate frame was presented for contouring control in two degrees of freedom systems. Task space is defined through set of curvilinear coordinates and systematic way for their design is given for two-dimensional case. This approach was used in [11] for two-loop control of biaxial servo systems. More detailed information about different methodologies adopted for contour tracking can be found in the literature reviews presented in [64] and [65].

Besides these generalized motion control requirements, application specific design is desired for most of the recent realizations of robotic laser processing systems [66], [67]. Integration of laser cutting process with robotic systems have opened many doors for various automated and manual operations. In terms of application fields, automated machining methodologies have found widespread usage in industrial applications [68], whereas semi-automated or manual machining techniques have been more frequently used in medical applications [69].

Among the medical applications of semi-automated robotic non-contact ma-

chining, researchers put particular attention on robot assisted laser surgery systems [70], [71]. Successful implementations on orthopedic surgery [72], laser scalping [73], bone cutting [74] and prosthetic surgery [75] have been illustrated in the literature by some authors. In [76] and [77] design, implementation and control of a novel robotic systems for assistive laser surgery are presented. Another computer controlled robotic laser surgery system is presented in [78]. The presented system has volume mapping function which works on the principle of cutting the volumetric information from an apriori stored data set (i.e. MR data) at certain planes and creating contours to track. This way, precise treatment of the area close to the boundary between tumors and normal brain tissue in neurosurgery is achieved. A similar approach was studied in [79] by navigated and model based calculation of ablation depth as a preoperative plan. A slightly different application is investigated in [80] by using computer guided scanning to improve the CO_2 laser assisted microincision precision. Further, utilization of flexible catheter systems are proposed in [81] and [82] to provide minimally invasive robotic laser surgery.

Integration of haptic feedback in the manipulation system for laser assisted surgery have recently been popularized by some researchers [83]. In [84], authors discuss a prototype system to synthesize haptic feedback through a robotic arm held by the operator, when the focal point of the laser is coincident with a real surface. Likewise, successful results about haptic feedback including cognitive laser surgery systems [85] and systems to compensate the tremor in laser surgery via haptic feedback [86], [87] are presented by some researchers. Methods for active guidance of operator motion through haptic feedback have been studied in [88] and [89] under the concept of semi-

automated laser surgery. Studies about utilization of haptic feedback in robot assisted laser cutting systems are supposed to extend the laser scalping abilities of the operator to augment the success rates in surgeries [90], [91]. For further discussion about use of robot assisted laser surgery systems and applications, the reader is addressed to the reviews presented in [92] and [93] and the literature summary given in [94].

Chapter III

3 Contour Extraction and Trajectory Generation Using an External Source

Machining of a certain shape by a laser unit requires the embedded contour information be extracted from its source prior to the actual operation. Traditionally, this contour information is stored in certain data format (like Drawing Exchange Format - .dxf) in the digital form depending on the environment or software where the original shape is generated. However, the shapes generated in the digital form can only contain certain combination of simple geometries (i.e. line or circle) which puts a limitation over the realization of random geometries. On the other hand, machining requirement of these arbitrary shapes arise in many applications including production of customized goods made by laser engraving (Figure 3.1-A) or laser microdissection of carcinogenic tissues in medicine (Figure 3.1-B).

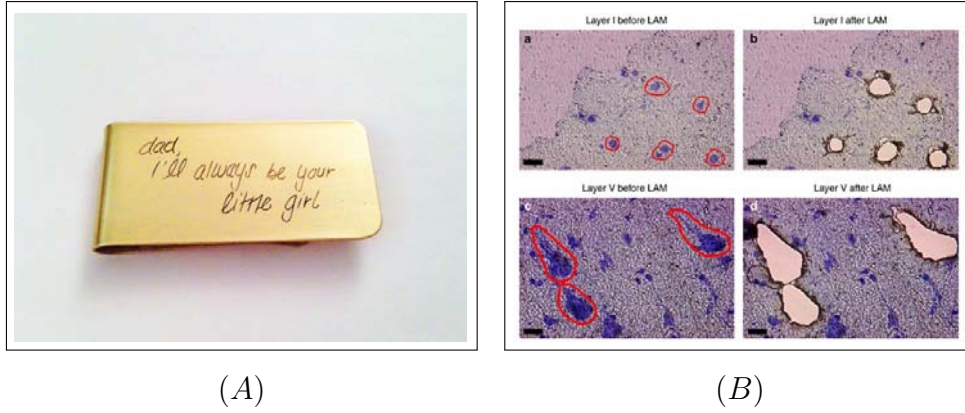


Figure 3.1: Random Shape Laser Machining Applications; Laser Engraving (A), Laser Microdissection (B)

3.1 Image Processing Based Trajectory Acquisition

Processing any arbitrary reference under laser micromachining unit can be possible via feeding the corresponding trajectory information from an external source to the system. For this purpose, image based reference generation is proposed in the following discussion. Making use of images as the main source of reference brings the advantage of easy application without requirement of complicated and expensive tools. Moreover, most of current state of the art surgery devices already include an inspection unit to provide the surgeon with visual feedback during operation. Hence, the developed methodology can easily be integrated to the existing machining systems. The general purpose of this section is to describe the intermediate steps to extract the shape information embedded in an image and to transform this information as a set of data points to the machining unit for generation of smooth motion trajectories. In that sense, the following image processing steps are carried out in the implementation phase;

- Image Enhancement and Edge Detection
- Determination of Initial Data Point
- Detection of Data Points
- Scaling to Operational Space
- Increasing Data Resolution

The details related to the content of each of these steps are explained in the following subsections.

3.1.1 Image Enhancement and Edge Detection

The shape information of a closed or open contour can simply be acquired from the edges of that contour in the image. Hence, the edge detection operation is of primary interest in order to definitely determine the contour boundaries. It is important to note here that robust detection of edges in an image is by itself a whole research field. The methodology adopted here is a simple application from the results of this field just to show the feasibility of the overall approach. Further analysis related to the robustness and correctness of detected edges is beyond the scope of this study. It is assumed here that in case of highly complicated image, the user is capable of manually drawing the contours over the image using either one of a tablet computer or a smart phone as shown in [95].

Prior to the extraction of information from an image, a valid assumption would be that the input image is corrupted with white noise. So, before continuing with edge detection, the input image is filtered with a Gaussian mask.

Further, the edges of the input figure are detected via the Canny [96] operator. Since there is always possibility of discontinuities, the detected edge content is passed over two other morphological image operations, namely, dilation and erosion. This step is usually required to guarantee the existence of a smooth and continuous curvature. Following the morphological operations, the edge content is obtained and converted to binary form for the next step of algorithm. Examples of the smoothed images and correspondingly detected edge contents are shown in Figure 3.2 below for open and closed contours respectively.

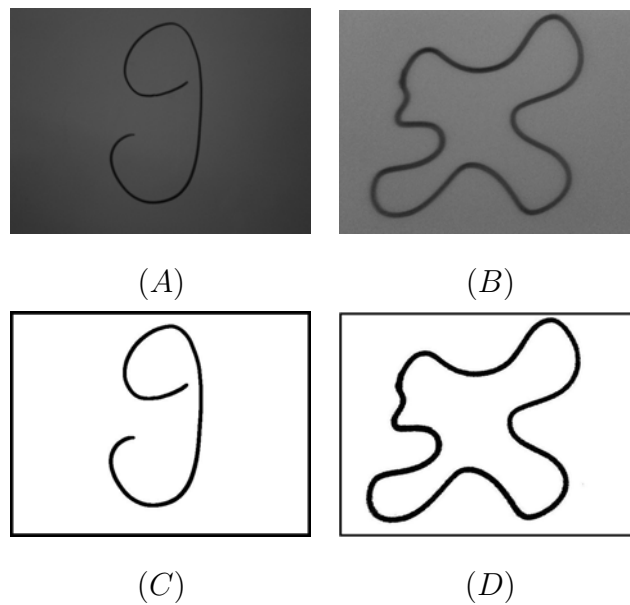


Figure 3.2: Examples of Smoothed Images and Detected Edges; Smooth Open Contour Image (A), Smooth Closed Contour Image (B), Open Contour Detected Edges (C), Closed Contour Detected Edges (D)

3.1.2 Determination of Initial Data Point

Successfully acquired edge content carries the richest information to generate the motion constraining trajectory. Once the edge detection is completed, the initial point of the trajectory has to be determined in order to initiate the point detection algorithm. For the determination of initial points different methodologies are adopted for open and closed contour images. For open contour images, the user is prompted to specify the first and last points over the figure since there is no prior information related to the starting and ending point of the open contour segment. On the other hand, for closed contours, this process can be made automatically since the whole contour has to be traced by the tracking algorithm. In that sense, for closed contour images, an algorithm scans the image in order to find the first point of edge that is grabbed within an evolving square. Since the shape has a closed structure, the first point detected by the algorithm can also be used as the last point to exit the data point detection algorithm described in the next section. The illustrations for open and closed contour initial point detections are shown below in Figure 3.3.

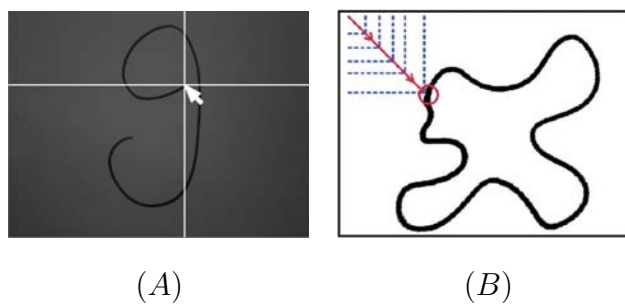


Figure 3.3: Illustration of Initial Data Point Selection; User Prompted Selection for Open Contour Images (A), Automatic Selection for Closed Contour Images (B)

3.1.3 Detection of Trajectory Data Points

Starting from the initial point determined in the previous section, a window travels over the curve in the tangent direction of image gradient until it reaches the last point of the contour. The gradient is calculated by a 2D first order derivative mask over the image using the following formulation [97];

$$I_x V_x + I_y V_y + I_t = 0 \quad (1)$$

where, $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $V_x = \frac{dx}{dt}$ and $V_y = \frac{dy}{dt}$ stand for the derivative of image in x-direction, derivative of image in y-direction, optical flow x-axis velocity and optical flow y-axis velocity respectively. In equation (1), the term $I_t = \frac{\partial I}{\partial t}$ represent the time derivative of image assuming a sequence of images exist in the system. In the context of this study, that sequence is iteratively generated by removing the pixels in the centroid of tracking window of previous image from the next image. Once the flow velocity ($[V_x, V_y]^T$) is determined, the center of the window is moved along that velocity direction to continue with the detection of next centroid point. In the next location of window, the new centroid, new gradient and new window location is calculated and this process is carried out iteratively until the last point of the contour is detected. During the motion, the collection of all centroid points of the window gives the order of points that defines the reference trajectory in image space. The procedure of detecting the data points and the detected point set is shown in Figure 3.4 for open and closed contours respectively.

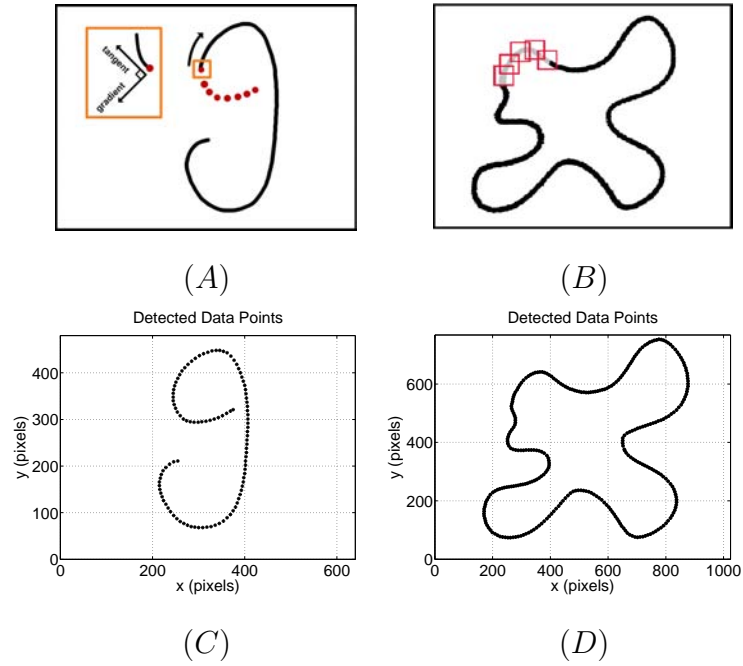


Figure 3.4: Data Point Detection; Illustration for Open Contour Image (A), Illustration for Closed Contour Image (B), Detected Data Points for Open Contour (C), Detected Data Points for Closed Contour (D)

3.1.4 Scaling to Operational Space

The coordinates of extracted data points are originally represented in terms of pixels since the centroid of the moving window is nothing but a single pixel. On the other hand, the operational space that should keep the trajectory information is consisted of cartesian coordinate system. Hence, the original pixel coordinate system should be transformed to the operational space coordinates. The conversion from image space to task space is carried

out through the following affine mapping;

$$\begin{bmatrix} x_O \\ y_O \end{bmatrix} = \begin{bmatrix} \left(\frac{O_W}{I_W}\right) & 0 \\ 0 & \left(\frac{O_H}{I_H}\right) \end{bmatrix} \begin{bmatrix} x_I \\ y_I \end{bmatrix} + \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} \quad (2)$$

where, $[x_O, y_O]^T$ and $[x_I, y_I]^T$ stand for the operational space coordinates (in meters) and pixel-wise image space coordinates of the constraining contour respectively. Here O_W , O_H , I_W and I_H represent the operational space width (i.e. x-axis), operational space height (i.e. y-axis), image space width and image space height respectively. The terms β_x and β_y in equation (2) are the x and y axis offset values and can be given as:

$$\beta_x = x_F \left(\frac{O_W}{I_W}\right) + \frac{O_W}{2} \quad (3)$$

$$\beta_y = y_F \left(\frac{O_H}{I_H}\right) + \frac{O_H}{2} \quad (4)$$

where, x_F and y_F are the first detected points on the constraining trajectory. Illustration of scaling process is given in Figure 3.5 below.

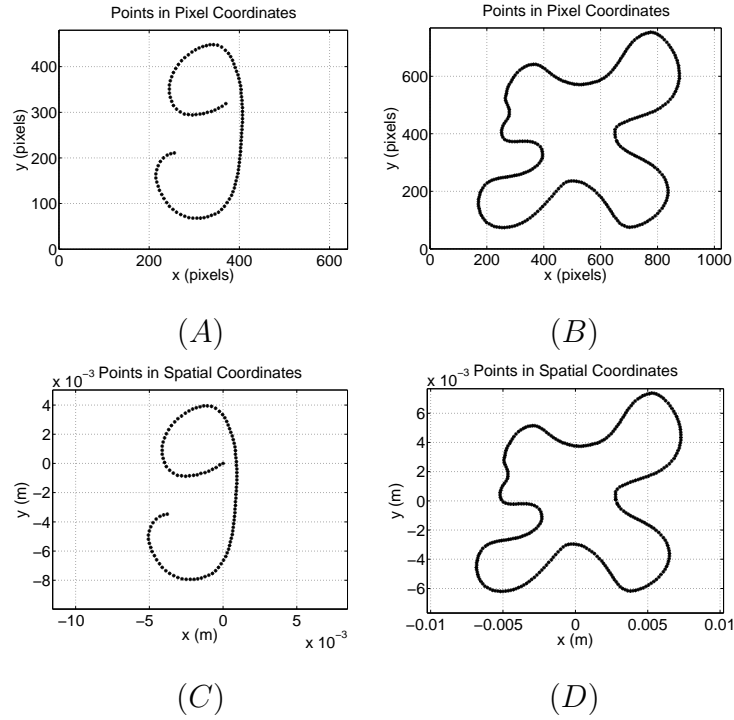


Figure 3.5: Scaling from Image to Operational Space; Open Contour Data Points in Image Space (A), Closed Contour Data Points in Image Space (B), Open Contour Data Points in Task Space (C), Closed Contour Data Points in Task Space (D)

3.1.5 Increasing Data Resolution

The task space contour data, originally acquired from a 2D image, usually do not contain enough number of data points to provide a smooth reference trajectory for the tracking controller. Hence, an intermediate operation would be needed to increase the total number of ordered data points and come up with a fine resolution trajectory. In order to preserve the shape of the given spatial curvature, a polynomial interpolation methodology is adopted in the context of this paper.

Given a set of N ordered data points in plane ($P_i = [x_i, y_i, z_i], i = 1, 2, \dots, N$), the data enhancement algorithm first distributes the set into arrays of points for three independent axes. Then, for all of the three axes, starting from the first point the algorithm takes every 3 consecutive points in a window and applies quadratic polynomial interpolation within the window. At every incremental motion of the window, an intermediate point is placed between the first two points based on the corresponding function. The interpolation window travels until the last data point is included. In the last position of the interpolation window, two intermediate points are inserted. First intermediate point is inserted between the first and second, and second intermediate point between the second and third essential points of the window. This process is handled in an iterative loop with iteration number k being set a priori.

Mathematically speaking, after one passing through the initial data set with the interpolation window, a total of $N - 1$ data points are added over the existing N data points. This means that after k iterations, the total number of data points in the system is increased to;

$$\#_{DataPoints} = 2^k (N - 1) + 1. \quad (5)$$

For the sake of completeness, results obtained from resolution enhancement procedure are given in Figure 3.6 below;

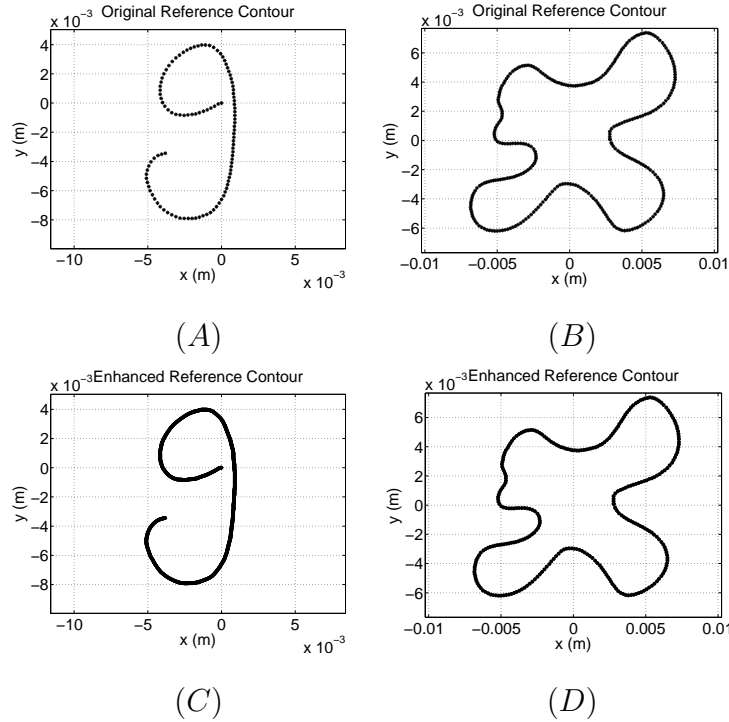


Figure 3.6: Increasing the Contour Resolution; Low Resolution Open Contour Data (A), Low Resolution Closed Contour Data (B), High Resolution Open Contour Data (C), High Resolution Closed Contour Data (D)

3.2 Generation of Time Dependent Trajectories

The shape information obtained via the resolution enhancement step described in the previous section contains nothing but a collection of data points with the corresponding x and y coordinates. In other words, the trajectory is obtained in the form $y(x)$. On the other hand, the robotic system which assumes these references will be operating based on trajectories that are functions of time ($x(t)$ and $y(t)$). Hence, as the second step of trajectory generation algorithm, here the methodologies used for transforming from $y(x)$ to time parameterized references $x(t)$ and $y(t)$ are analyzed. The analysis is

made differently for open and closed contour references. The details related to the approaches adopted for open and closed contour trajectory generation are given in the following two subsections respectively.

3.2.1 Open Contour Time-Dependent Trajectory Generation

The trajectory generation algorithm for open contour references receives the ordered set of data points acquired from the image and augmented in magnitude by means of intermediate point insertion through quadratic polynomial interpolation. On the other hand, in order to generate certain trajectory, one has to parameterize these ordered data points with respect to time. Speaking about parameterization, one can think of fitting a mathematical model over the given shape of open contour. However, this (open shape contour modeling) is by itself a whole research problem being studied by many researchers under the topics of active contours, particularly referred as snakes [98], [99]. Moreover, since most of the proposed solutions work based on energy minimization schemes to fit segmented polynomials that have relatively low degree, yet there is not a robust and easy to use solution from these studies. Besides these difficulties, one other weakness of snake based methods is that they cannot be used to model curves that can have self crossing at certain points. In other words, the snake based methods fail to model the shape if the $y(x)$ trajectory crosses itself at some points, resulting in a combination of open and closed contours. Having considered these problems, rather than adopting a modeling based approach, in the context of this study, an algorithmic approach is adopted for the generation of reference trajectories based on time parameterization. Below the details of this algorithm is explained; Having acquired a fine resolution trajectory of the given reference curve, one

can insert the tangential velocity constraint $V_T(t)$ over the system. Considering a real-time control system, apriori known velocity constraint means that the reference tangential velocity ($V_T[kT_s]$) is known for each sample kT of the real time algorithm. Making use of this information, one can integrate the tangential velocity reference to get the desired distance $R[kT_s]$ to be traveled at sample k of the system running in real time as follows;

$$R[kT_s] = V_T[kT_s]T_s \quad (6)$$

where, T_s stand for the sampling period of the real time algorithm. Furthermore, the distance between any two points $P_i = [x_i, y_i]$ and $P_j = [x_j, y_j]$ over the reference trajectory is given by

$$D_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (7)$$

Since there is a finite number of data samples, making direct use of the distances between points on the trajectory, one can only acquire segments with certain error bounds that are determined by the resolution of enhanced data. The results obtained using such an approach have been shown to perform successfully in construction of the constant velocity trajectory profile [100]. However, further improvement of precision is possible if equation (7) can be implicitly solved. Below the details related to this approach are presented.

The algorithm adopted in this study works in an iterative loop going over the entire enhanced data set of the reference trajectory. Starting from the first two points, the distance between the given points is calculated based on the formula given in equation (7). If the calculated distance is smaller than the

reference distance $R[kT_s]$, the second point is incremented by one to calculate the new distance. This procedure is executed until the calculated distance becomes greater than the reference distance $R[kT_s]$. Once the first point that results with bigger distance (i.e. P_k) is acquired, then a linear interpolation is made between the last two points P_k and P_{k-1} over which at least one exact solution of equation (7) exists for distance $R[kT_s]$. The intermediate point P_* with exact solution is recorded as the second reference position point in the constant velocity reference trajectory. The algorithm continues until the last point in the trajectory is obtained. A schematic illustration of this process is given in Figure 3.7 below while results of the time parameterization process for open contour trajectory generation algorithm is illustrated in Figure 3.8.

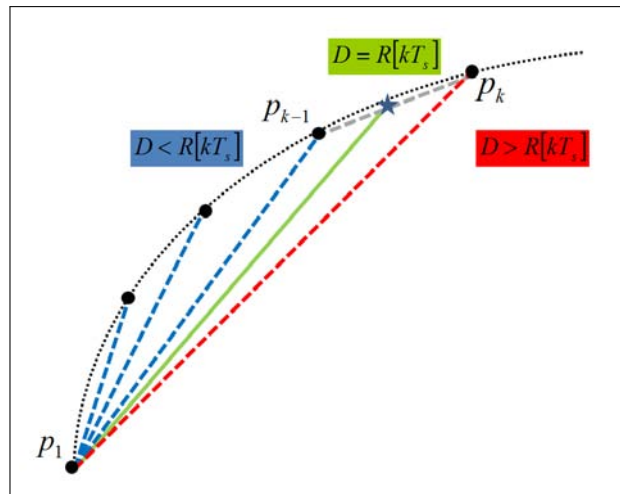


Figure 3.7: Schematic Illustration of Interpolation for Velocity Constrained Segmentation

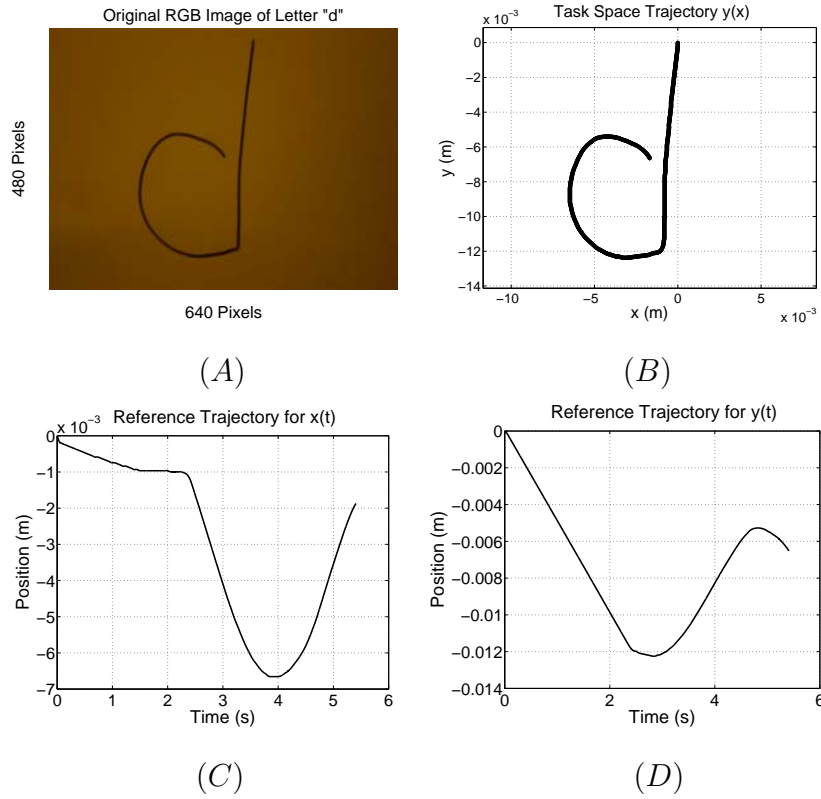


Figure 3.8: Reference Trajectory Generation for Task-Space Coordinates. Original Image (A), Extracted and Enhanced Reference Trajectory Data Points (B), Time-Based X-Coordinate Reference Trajectory for Constant Tangential Velocity (C), Time-Based Y-Coordinate Reference Trajectory for Constant Tangential Velocity (D)

3.2.2 Closed Contour Time-Dependent Trajectory Generation

In closed contour shapes, benefitting from the self-repeating structure of the curvature, possibility of parameterization with periodic functions arise. In that sense, the second segment of the trajectory generation algorithm is reserved for the discussion about fitting a smooth and parametric curve to the closed contour obtained via the algorithm described in Trajectory Acquisition

section. In the context of this study, we adopted the use of Elliptic Fourier Descriptors (EFDs) for the representation of given closed curvature.

In recent years, EFDs have been popularized and studied widely for the parametric representation of 2D closed curves [101], [102], [103] with further extensions on 3D geometries [104]. The major ease brought by EFDs is the ability to represent the closed curve with a finite set of parameters which are obtained via an ordered combination of sinusoidal functions with different frequencies. Moreover, the advantage that the shape information is kept in the low frequency components makes EFDs further feasible for application. The n -harmonic Elliptic Fourier Descriptor representation of any 2D curve can be given as,

$$x(\theta) = a_0 + \sum_{k=1}^n \{a_k \cos(k\theta) + b_k \sin(k\theta)\} \quad (8)$$

$$y(\theta) = c_0 + \sum_{k=1}^n \{c_k \cos(k\theta) + d_k \sin(k\theta)\} \quad (9)$$

where, a_0 and c_0 is the center location of the curve and a_k , b_k , c_k and d_k ($k = 1, \dots, n$) are the Elliptic Fourier coefficients of the 2D curve up to n^{th} harmonics. In equation (8) and (9), θ stands for the parameterization variable of the represented shape. This way, the curve preserves its unique shape for $\theta \in [0, 2\pi]$ and repeats with a period of 2π .

Given a set of M points from an image (i.e. locations of M points that lie on the curve boundary), one can make use of $2M$ number of data to calculate the $4n + 2$ coefficients that represent the corresponding curve. Usually in practice one cannot guarantee the equivalence of number of data points and the coefficients (i.e. usually we have $M \neq 2n + 1$). Hence, for practical

purposes, least squares approximation to minimize a quadratic error between the estimated points and the actual points. Explicit formulas for calculation of coefficients a_k , b_k , c_k and d_k ($k = 0, 1, \dots, n$) is given in [103]. The results obtained from EFD representation is depicted in Figure 3.9 with different number of harmonics.

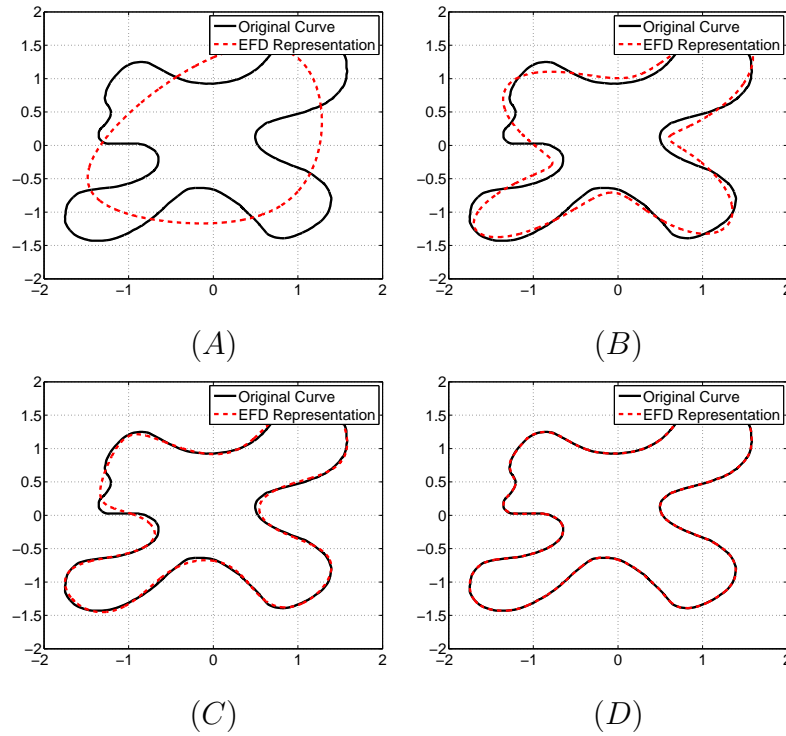


Figure 3.9: Representation of Closed Contour Trajectory $y(x)$ with EFDs. Number of Harmonics; 2 (A), 5 (B), 10 (C), 30 (D)

The advantage brought by adopting a parameterized curve lies in the possibility of easier time parameterization for $x(t)$ and $y(t)$ from $y(x)$. In order to move from the given 2D shape (parameterized with respect to θ) to the time parameterized curve, a similar methodology is adopted like the one used for open contour curves. However, benefitting from the parametric represen-

tation, this time the number of data points on the actual trajectory can be made arbitrarily large by taking infinitesimal changes $\delta\theta$. Following this routine, the algorithm implemented for the open contour time dependent trajectory generation is also adopted over the closed contour shapes. Below, an example of time parameterization for a closed contour shape is given in Figure 3.10. In order to illustrate the application flexibility of this approach, the original contour data of this figure is generated in a CAD program and saved in "JPEG" format to be ready for the image processing algorithm. In this example, the constant tangential velocity reference is taken as $0.05m/s$ during the generation of time-parameterized trajectories.

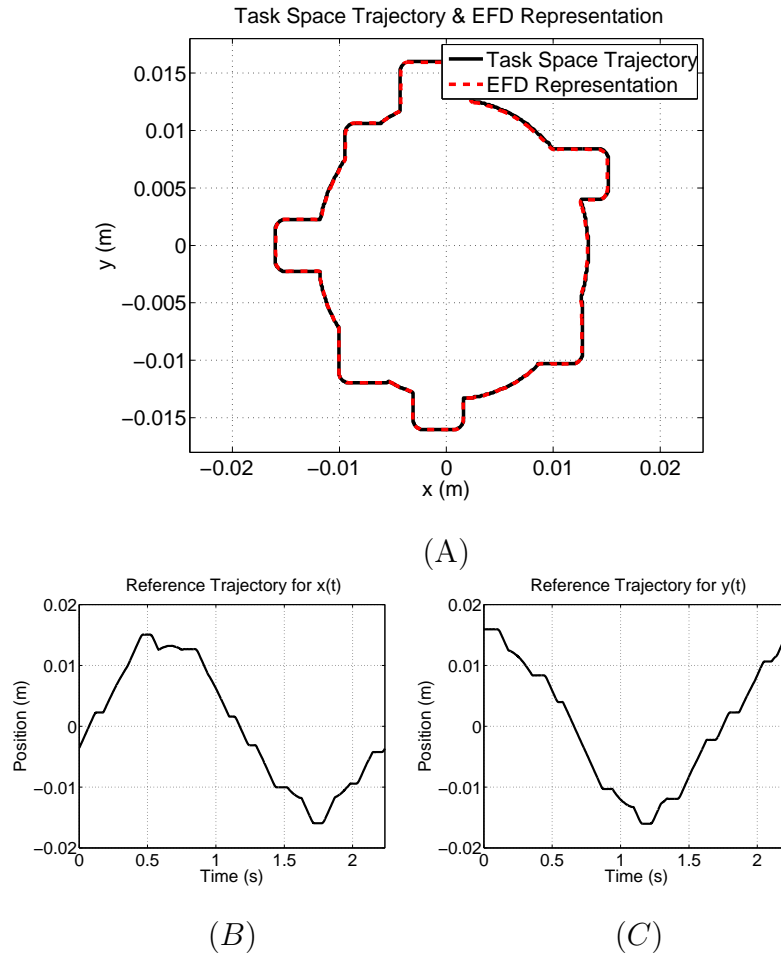


Figure 3.10: Time Parameterization of Closed Contour Shape. Original Task Space Trajectory and Representation with EFDs (A), Time-Based X-Coordinate Reference Trajectory for Constant Tangential Velocity (B), Time-Based Y-Coordinate Reference Trajectory for Constant Tangential Velocity (C)

3.2.3 Mapping to Joint Space Coordinates

Having obtained reference trajectories $x(t)$ and $y(t)$, the next step in the realization of the time dependent reference trajectory generation process is

mapping these references from task space to joint space through the inverse kinematics of the positioning system. Under the condition that the robotic system used for the realization is fully actuated, there exists one to one mapping between these two spaces which can be given by;

$$\mathbf{q} = \mathcal{I}(\mathbf{x}) \quad (10)$$

where, $\mathbf{q} = [q_1(t), q_2(t), q_3(t)]^T$, $\mathbf{x} = [x(t), y(t), z(t)]^T$ and $\mathcal{I}(\cdot)$ stand for the joint space references, task space references and the inverse kinematics operation respectively. Due to the requirement of planar processing in non-contact machining, the task space reference $z(t)$ can be taken as a constant value. The redundancy introduced by the extra degrees of freedom (i.e. z-axis) in the system can also be used for the formation of intermediate joints in an application specific way. An example of such a system is analyzed in Chapter 5 with a 3-DOF system where the process is handled by constraining the motion in the third dimension. With the mapping from task to joint space coordinates, it becomes possible to derive a controller that can work in the configuration space while performing the tracking of desired arbitrary trajectory provided by the image data with desired tangential velocity reference. For the sake of completeness, below in Figure 3.11 an illustration of generated joint space reference trajectories is given for the reference task space trajectories shown in Figure 3.8 (C) and (D). These trajectories are generated for a 3-DOF linear delta mechanism analyzed as the experimental platform used for realization of algorithms in Chapter 7.

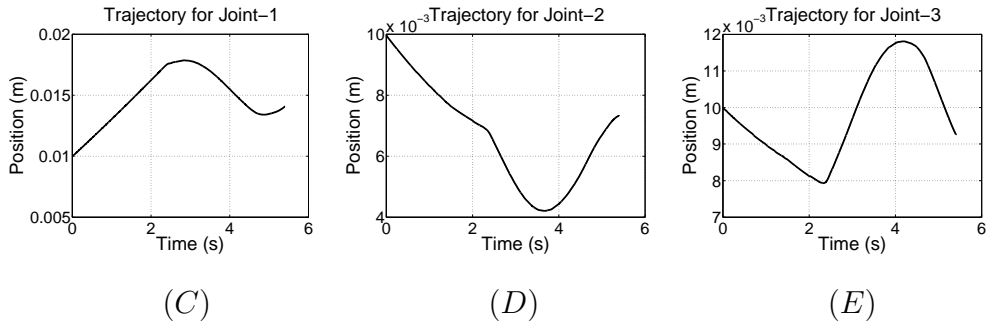
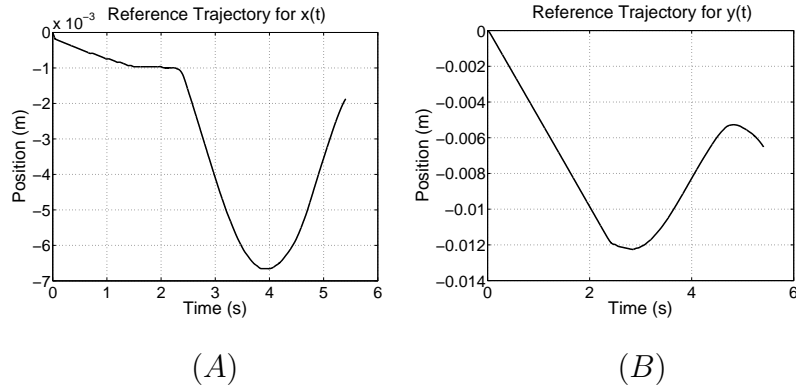


Figure 3.11: Mapping from Operational Space Coordinates to Configuration Space Coordinates. X-Coordinate Reference Trajectory (A), Y-Coordinate Reference Trajectory (B), Joint-1 Reference Trajectory (C), Joint-2 Reference Trajectory (D), Joint-3 Reference Trajectory (E)

3.2.4 Time Based Spline Approximation

Following the acquisition of joint space position trajectories, the corresponding velocity and acceleration references should be generated so that they can be used in the controller. Unfortunately, the reference trajectory for the joint angles, alone, is not sufficient to produce smooth velocity and acceleration reference profiles. In order to assure perfect replication of the shape specified either in the image or in a technical drawing and achieve high quality machining, existence of smooth velocity and acceleration references is a nec-

essary condition. Jerky motion and position overshoot at the corner points should be minimized or completely avoided in order to guaranty high quality machining of any shape.

In order to have smooth profiles for velocity and acceleration references, the time based spline interpolation method [41, 105, 106] is implemented. The spline approximation is carried out on the time based configuration space trajectory data in the configuration space. The basic logic behind spline interpolation relies on fitting polynomials of order three over every four consecutive data point of each joint reference angle. This way, the position references can be parameterized with a continuous and differentiable curve. Hence, using direct differentiation, one can also obtain the smooth velocity and acceleration references of the trajectory as functions of time. As the result of the spline interpolation, polynomials that represent the position, velocity, acceleration and jerk for each sampling period are obtained in the following forms;

$$q_i(t) = a_i t^3 + b_i t^2 + c_i t + d_i \quad (11)$$

$$\dot{q}_i(t) = 3a_i t^2 + 2b_i t + c_i \quad (12)$$

$$\ddot{q}_i(t) = 6a_i t + 2b_i \quad (13)$$

$$\dddot{q}_i(t) = 6a_i \quad (14)$$

where, $i = 1, 2, 3$ for a 3-DOF manipulator and t stands for the time increment in the corresponding sample. Since the coefficients ($a_i, b_i, c_i, d_i, i = 1, 2, 3$) of these polynomials are calculated through the interpolation process at the position level, one can substitute those coefficients for each sampling period, obtaining the corresponding velocity and acceleration com-

mands.

3.3 Generation of Time Independent Trajectories

The time dependent trajectories are required for machining operations that are automatically done by a robotic system. On the other hand, there might be situations where the reference trajectory is generated without being parameterized with respect to time. The main objective of generating time independent parameterization of some given contour can be reflected in assistive machining applications. Ability of manual intervention (i.e. operator assistance) during laser cutting process is of particular importance especially for medical operations like laser surgery [67], [107]. In such systems, the robotic system can provide assistance to the surgeon by confining the end effector motion within certain path in which free motion can be executed.

Constraining certain pre-determined path requires the parameterization of the corresponding trajectory in a time-independent manner. Originating from this point, in the discussion given below, a methodological way of time independent trajectory (also referred as the motion constraint) generation is presented based on segmented circular constraints. Such an approach was discussed in [108] under the name "*free form path generation*", where the authors have proven mathematically that circular constraints give less contouring error than linear constraints. As in the case of time dependent contours, the main source of information for the generation of time independent trajectories is again obtained from the set of operational space data points acquired from an image. The details related to the generation of motion constraining paths using these data points is described in the following subsections.

3.3.1 Selection of Constraint Generating Data Set

The basics of this algorithm relies on iteratively generation of constraints passing over the closest data points to the point where the end effector of robotic system sits. In other words, at every iteration of the algorithm, the array of distances between the robot end effector coordinates and the operational space coordinates of trajectory data points is calculated. Over that distance array, the trajectory point that has the minimum value is selected as the mid point of constraint data array.

Let us call that distance minimizing data point as $\mathbf{P}_m = [x_m, y_m]^T$ and the point where end effector of the robotic system sits as $\mathbf{P}_e = [x_e, y_e]^T$. In order to generate the circle, at least three points are required. However, there is possibility that these three points can lie on the same line bringing the problem to an ill-posed situation. In order to prevent this situation and have a well-posed problem, the closest $2k$ points are taken into consideration and the following data set $\mathbf{D} \in \mathbb{R}^{(2k+1) \times 2}$ is generated for circle construction segment of the algorithm described in the next subsection.

$$\mathbf{D} = [\mathbf{P}_{m-k}, \dots, \mathbf{P}_m, \dots, \mathbf{P}_{m+k}]^T \quad (15)$$

3.3.2 Generation of Circular Constraints

Once the data set of points to generate the constraints is acquired, fitting of a circle over these points is carried out using a least squares based approach. In the most general form, the equation of a circle in plane centered at point $[c_x, c_y]^T$ and passing over point $[x_i, y_i]^T$ with radius r can be given as;

$$(x_i - c_x)^2 + (y_i - c_y)^2 = r^2 \quad (16)$$

Expanding equation (16), one can formulate the same circle equation as follows;

$$\eta_1 x_i + \eta_2 y_i + \eta_3 = -(x_i^2 + y_i^2) \quad (17)$$

where, $\eta_1 = -2c_x$, $\eta_2 = -2c_y$ and $\eta_3 = c_x^2 + c_y^2 - r^2$. For a total of $2k + 1$ data points, equation (17) can be recast in the form of a linear regression problem as follows;

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_{2k+1} & y_{2k+1} & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} = \begin{bmatrix} -(x_1^2 + y_1^2) \\ -(x_2^2 + y_2^2) \\ \vdots \\ -(x_{2k+1}^2 + y_{2k+1}^2) \end{bmatrix} \quad (18)$$

Equation (18) is of the form $\boldsymbol{\varphi}\boldsymbol{\theta} = \boldsymbol{\mu}$ and is linear in parameters with an overdetermined structure. The solution of this equation comes with a least squares based approach as follows;

$$\boldsymbol{\theta} = \boldsymbol{\varphi}^\# \boldsymbol{\mu} \quad (19)$$

where, $\boldsymbol{\varphi}^\# \in \mathbb{R}^{3 \times (2k+1)}$ stand for the right weighted pseudo inverse of the matrix $\boldsymbol{\varphi}$. Having acquired the parameters to generate a circle, the constraint that has to be satisfied for the motion of robotic system can be realized. The details on realization of constraints are provided in Appendix A. A depiction of constraints generated over the system is given in Figure 3.12 below.

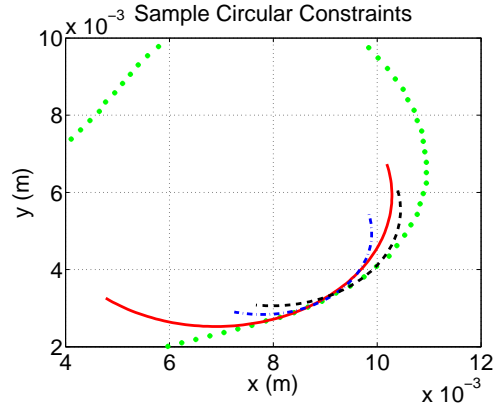


Figure 3.12: Representation of Time Independent Trajectory Generation Methodology: Illustration of Motion Constraining Circles

For the sake of completeness, an algorithmic summary of time independent trajectory (i.e. constraint) generation procedure is presented in Algorithm 1.

Algorithm 1 Summary of Constraint Generation Procedure

Given the set of data points \mathbf{P}_j for $j = 1, \dots, N$;

- Calculate the distance L_j between \mathbf{P}_e and \mathbf{P}_j for $j = 1, \dots, N$;
- Find the minimum distance point; $\mathbf{P}^* = \arg \min L_j$
- Select constraint generating data set around $\mathbf{P}^* = \mathbf{P}_m$

$$\mathbf{D} = [\mathbf{P}_{m-k}, \dots, \mathbf{P}_m, \dots, \mathbf{P}_{m+k}]^T$$

- Form corresponding circle equation
 - Obtain parameters η_1 , η_2 and η_3
 - Apply motion constraining controller
-

Chapter IV

4 Motion Control for Automated Non-Contact Machining

In non-contact machining, depth of the processed feature is dependent on the properties of material and the corresponding machining unit. Usually, the interaction between the sample to be processed and a laser cutting unit can be modeled with certain assumptions in the background [109], [110]. In most of the applications, the removed material from the processed region is inversely proportional to the processing velocity of the corresponding robotic system. Under the valid assumption that non-contact machining unit preserves a uniform operational structure (i.e. constant pulse repetition for a laser), the machining precision depends on the velocity tracking performance of robotic system's controller. This interdependence, therefore, requires a precise tracking of pre-defined velocity profile by the machining system. In regions where actual velocity is different from the average velocity that is required to be maintained, over- or under- ablation occurs. This situation is observed at regions where it is difficult to maintain the desired velocity profile such as sharp turns, where the relative motion between the machining unit and the workpiece is either accelerating or decelerating. Therefore, different control techniques need to be employed when dealing with complex

and random trajectories.

In this chapter, the derivation of the controllers used for the realization of automated non-contact machining is presented. Due to the structural differences between open and closed contour references, two different controller schemes are proposed for the tracking of given time dependent trajectories. Throughout the derivation given in the following sections, it is assumed that the implementation systems in charge are consisted of multi-DOF robotic platforms. Hence, prior to the derivation of controllers, background information is presented about the dynamics of multi-DOF motion control systems. Moreover, since both of the controllers are derived based on the assumption of acceleration tracking, a brief summary of robust acceleration control framework for multi-DOF systems is also presented in the preliminary discussion.

4.1 Background Information

4.1.1 Multi-DOF System Dynamics

For an n DOF fully actuated motion control system, general form of the configuration space dynamics can be given with the following equation;

$$\mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{T}^{\text{ref}} - \mathbf{T}^{\text{ext}} \quad (20)$$

where, $\mathbf{q} \in \mathbb{R}^{n \times 1}$, $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{n \times n}$, $\mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times 1}$, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ and $\mathbf{T} \in \mathbb{R}^{n \times 1}$ stand for the vector of generalized coordinates, kinetic energy (i.e. inertia) matrix, vector of Coriolis, centripetal and viscous forces, vector of forces induced by gravity and the vector of forces acting directly on the joints respectively. Furthermore, the superscripts $^{\text{ref}}$ and $^{\text{ext}}$ represent the reference

and external variables.

The dynamics given in (20) contain the effect of all coupling forces due to the mechanical constraints enforced by the kinematic chain of the system under consideration. In order to have robust motion control, this nonlinear and coupled system can be decoupled and linearized around a nominal operating point. In that sense, the dynamics can be rearranged as follows;

$$\mathbf{A}_n \ddot{\mathbf{q}} = \mathbf{K}_n \mathbf{i}^{\text{ref}} - \mathbf{T}^{\text{dis}} \quad (21)$$

$$\mathbf{T}^{\text{dis}} = \{ \Delta \mathbf{K} \mathbf{i}^{\text{ref}} + \Delta \mathbf{A}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) + \mathbf{T}^{\text{ext}} \} \quad (22)$$

where, $\mathbf{A}_n \in \mathbb{R}^{n \times n}$, $\mathbf{K}_n \in \mathbb{R}^{n \times n}$ and $\mathbf{i}^{\text{ref}} \in \mathbb{R}^{n \times 1}$ stand for the diagonal constant parameter matrices of nominal inertia and torque constants and the vector of input currents. In equation (21), the term $\mathbf{T}^{\text{dis}} \in \mathbb{R}^{n \times 1}$ represent the combination of all undesired torques (i.e. disturbances) that have potential to divert the motion from the desired reference. The content of disturbance torque is given in equation (22). Here, $\Delta \mathbf{K} \in \mathbb{R}^{n \times n}$ and $\Delta \mathbf{A}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ are the matrices keeping the fluctuations of torque constant and inertia from their corresponding nominal values. These matrices can basically be interpreted as the components responsible for current and acceleration induced torques in the system respectively. Moreover, in equation (22), the term $\mathbf{T}^{\text{ext}} \in \mathbb{R}^{n \times 1}$ lumps all other non-modeled external torques acting on the system as additional disturbance.

4.1.2 Disturbance Observer & Robust Acceleration Control

Estimation of the disturbances in (21) can be handled by the incorporation of a Disturbance Observer (DOB) using the assumed nominal parameters of the system and low-pass filters as explained in [111]. Mathematically, this estimation can be represented as follows;

$$\hat{\mathbf{T}}^{\text{dis}} = (\mathbf{K}_n \mathbf{i}^{\text{ref}} - \mathbf{A}_n \ddot{\mathbf{q}}) \mathbf{H} \quad (23)$$

where, $\mathbf{H} \in \mathbb{R}^{n \times n}$ is the diagonal matrix whose components are low-pass filters of structure $h_{ii} = g_i / (s + g_i)$ for $i = 1, \dots, n$. In order to enforce robust motion control in configuration space, the estimated disturbance in (23) can be inserted back to (21). This way, the dynamics of the overall system takes the following form;

$$\mathbf{K}_n \mathbf{i}^{\text{ref}} = \mathbf{A}_n \ddot{\mathbf{q}} + \delta \mathbf{T}^{\text{dis}} \quad (24)$$

Here, the term $\delta \mathbf{T}^{\text{dis}} \in \mathbb{R}^{n \times 1}$ represents the remaining (i.e. uncompensated) disturbance still acting on the system. The magnitude $\|\delta \mathbf{T}^{\text{dis}}\|$ shows direct correspondence with the selection of disturbance observer filter gains g_i as analyzed in [112]. Selection of the highest possible values for these filter gains would enforce $\|\delta \mathbf{T}^{\text{dis}}\| \approx \mathbf{0}$ leaving the system in acceleration controllable form. Following this routine, the derivation of controller will be made to generate the desired accelerations for the joints. The structure of DOB for a multi-DOF system is depicted below in Figure 4.1.

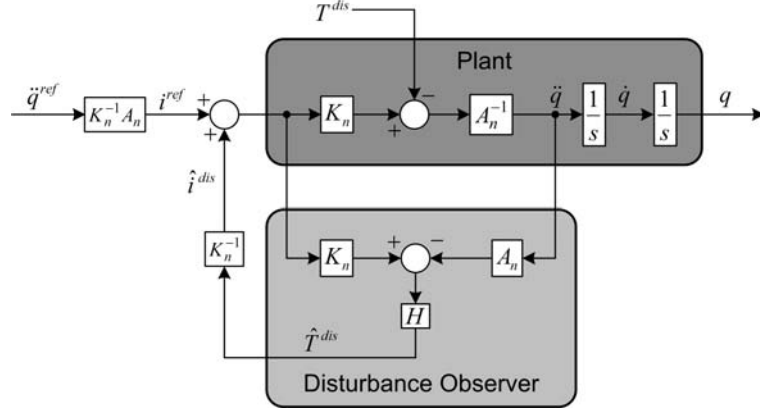


Figure 4.1: Generalized Structure of DOB for a Multi-DOF System

4.1.3 Mapping Between Operational and Configuration Spaces

For the system described by equation (24), operational space coordinates $\mathbf{x} \in \mathbb{R}^{n \times 1}$ can be represented using the relationship defined by the kinematic chain of the system under consideration. Mathematically speaking, the transformations between configuration space and operational space coordinates can be represented with the following set of equations;

$$\mathbf{x} = \mathcal{F}(\mathbf{q}) \quad (25)$$

$$\mathbf{q} = \mathcal{I}(\mathbf{x}) \quad (26)$$

where, $\mathcal{F}(\cdot)$ and $\mathcal{I}(\cdot)$ stand for the operators of forward and inverse kinematics respectively. In order to obtain the transformation between operational space and configuration space velocities and accelerations, one can differentiate the relationship given in equation (25) as follows;

$$\dot{\mathbf{x}} = \mathbf{J}_x \dot{\mathbf{q}} \quad (27)$$

$$\ddot{\mathbf{x}} = \mathbf{J}_x \ddot{\mathbf{q}} + \dot{\mathbf{J}}_x \dot{\mathbf{q}} \quad (28)$$

Here, $\mathbf{J}_x = \mathbf{J}_x(\mathbf{q}) = (\partial \mathcal{F}(\mathbf{q}) / \partial \mathbf{q})$ is the jacobian matrix associated with the tasks. In the context of this study, $\mathbf{J}_x \in \mathbb{R}^{n \times n}$ is assumed to be non-singular since the manipulators used in the experiments are mechanically limited to a such workspace.

4.2 Machining of Open Contour Trajectories

Successful accomplishment of machining for open contour trajectories relies on high performance tracking of operational space reference trajectories that are mapped to the joint space using inverse kinematics. This requires precise tracking of the time dependent trajectory references generated in equations (11) to (13). Robust and high precision tracking of these references can be realized using the acceleration control framework for which the prior background is provided in the preceding section. We can start the controller derivation by defining the tracking error as a linear combination of position and velocity errors;

$$\boldsymbol{\varepsilon} = \mathbf{S}_{p_2} (\dot{\mathbf{q}}^{\text{ref}} - \dot{\mathbf{q}}^{\text{res}}) + \mathbf{S}_{p_1} (\mathbf{q}^{\text{ref}} - \mathbf{q}^{\text{res}}) \quad (29)$$

where, $\mathbf{q}^\diamond = [q_1^\diamond, q_2^\diamond]^T$ and $\dot{\mathbf{q}}^\diamond = [\dot{q}_1^\diamond, \dot{q}_2^\diamond]^T$ stand for the position and velocity variables with $\diamond = \{\text{ref}, \text{res}\}$ representing the reference input and response obtained from the system respectively. In equation (29), the scaling matrices $\mathbf{S}_{p_1} = \text{diag}\{s_1\} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{S}_{p_2} = \text{diag}\{s_2\} \in \mathbb{R}^{3 \times 3}$ are diagonal matrices with positive entries (i.e. $s_1, s_2 > 0$) determining the respective weight of velocity and position over the generalized error. Enforcement of convergence

to zero value for this error can be possible with the following exponentially decaying error dynamics;

$$\dot{\boldsymbol{\varepsilon}} + \boldsymbol{\lambda}\boldsymbol{\varepsilon} = \mathbf{0} \quad (30)$$

where, $\boldsymbol{\lambda} = \text{diag}\{\lambda\} \in \mathbb{R}^{3 \times 3}$ is again diagonal matrix with $\lambda > 0$ determining the convergence rate of the error to zero. Substituting the content of error from equation (29) to (30), one can get;

$$\mathbf{S}_{\mathcal{P}_2} (\ddot{\mathbf{q}}^{\text{ref}} - \ddot{\mathbf{q}}^{\text{res}}) + (\mathbf{S}_{\mathcal{P}_1} + \boldsymbol{\lambda}\mathbf{S}_{\mathcal{P}_2}) (\dot{\mathbf{q}}^{\text{ref}} - \dot{\mathbf{q}}^{\text{res}}) + \boldsymbol{\lambda}\mathbf{S}_{\mathcal{P}_1} (\mathbf{q}^{\text{ref}} - \mathbf{q}^{\text{res}}) = \mathbf{0} \quad (31)$$

Equation (31) represents the configuration space dynamics of system under exponential convergence of error (30) to zero value. In order to satisfy this dynamics, the corresponding acceleration reference should be given to an acceleration controllable plant. Rearranging this equation, one can obtain the desired acceleration $\ddot{\mathbf{q}}^{\text{des}}$ to track the given time parameterized references as follows;

$$\ddot{\mathbf{q}}^{\text{des}} = \ddot{\mathbf{q}}^{\text{ref}} + \mathbf{K}_{\mathcal{P}_2} (\dot{\mathbf{q}}^{\text{ref}} - \dot{\mathbf{q}}^{\text{res}}) + \mathbf{K}_{\mathcal{P}_1} (\mathbf{q}^{\text{ref}} - \mathbf{q}^{\text{res}}) \quad (32)$$

where, $\mathbf{K}_{\mathcal{P}_2} \in \mathbb{R}^{3 \times 3}$ and $\mathbf{K}_{\mathcal{P}_1} \in \mathbb{R}^{3 \times 3}$ stand for the relative gains of controller for velocities and positions and have the following content;

$$\mathbf{K}_{\mathcal{P}_2} = \mathbf{S}_{\mathcal{P}_2}^{-1} (\mathbf{S}_{\mathcal{P}_1} + \boldsymbol{\lambda}\mathbf{S}_{\mathcal{P}_2}) \quad (33)$$

$$\mathbf{K}_{\mathcal{P}_1} = \mathbf{S}_{\mathcal{P}_2}^{-1} \boldsymbol{\lambda}\mathbf{S}_{\mathcal{P}_1} \quad (34)$$

The controller given in (32) has exactly the same structure with PD controller applied over a disturbance compensated plant. By selecting a relatively high

value in exponential decaying constant λ , the desired acceleration given in (32) will perform successful tracking of the given reference trajectory. Finite time convergence could also be possible by taking the error dynamics as;

$$\dot{\boldsymbol{\varepsilon}} + \lambda\boldsymbol{\varepsilon} + \mu\text{sign}(\boldsymbol{\varepsilon}) = \mathbf{0} \quad (35)$$

with $\mu > \mathbf{0}$. However, the sliding mode regime is prone to chattering errors which may impose loss of precision in the system. On the other hand, further enhancement of precision is possible by the utilization of preview control structure, since the reference velocity and acceleration values are known apriori for every data sample of the real-time algorithm.

4.2.1 Preview Controller

Preview control [41], [105] is used in applications where future information about reference data is already known. It works on the idea of feedforward compensation under the assumption that system is capable of tracking the acceleration references. As inputs to the controller; the current position reference command, one sampling step forward value of velocity reference command and two sampling steps forward value of the acceleration reference command are supplied. In [113], it is mathematically proven that control input structured in this way performs better in minimizing a quadratic functional of tracking error in the presence of exogenous inputs (i.e. uncompensated disturbances $\delta\mathbf{T}^{\text{dis}}$). Here, again the command values for the velocity and acceleration are obtained via direct differentiation of the parameterized position reference as shown in (12) and (13). Within the controller, each of these command data are multiplied by their respective control gains as de-

rived in (32) and summed to provide the desired reference command, which is then converted to current command to be supplied to the plant. The overall structure of the preview controller following the derivation given above is illustrated in the block diagram in Figure 4.2.

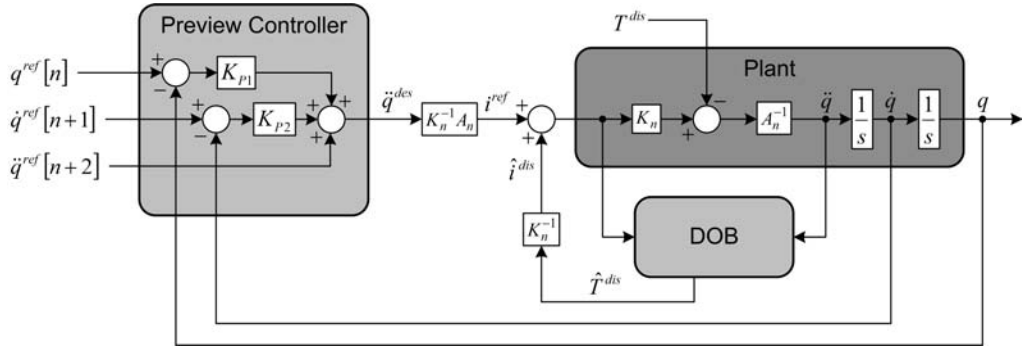


Figure 4.2: Structure of the Preview Controller

4.3 Machining of Closed Contour Trajectories

The closed contour shapes are structurally different than that of the open contours since they can be mathematically represented in closed form using the EFD based approach presented in the preceding chapter. Benefitting from this closed form representation, one can generate the desired shape of curvature very precisely and smoothly taking into consideration the velocity tracking requirement. Existence of smooth reference for the corresponding $x(t)$ and $y(t)$ trajectories enables the possibility to calculate the third order derivative as presented in equation (14). Smooth information about this third order derivative is important since it is used in the derivation of the contouring controller presented in the following analysis.

The problem of the designing controller for the compensation of contour tracking error can be formulated taking into consideration the satisfaction of

desired requirements of the task. These requirements are basically preserving the desired velocity profile in the direction tangent to the contour and enforcing zero velocity in the direction normal to the contour. This means that the error is defined in a space (referred as "orthogonal space" in the rest of the text) where the contour profile is dynamically transformed into its tangential and normal components. An illustration of this new coordinate system is given below in Figure 4.3.

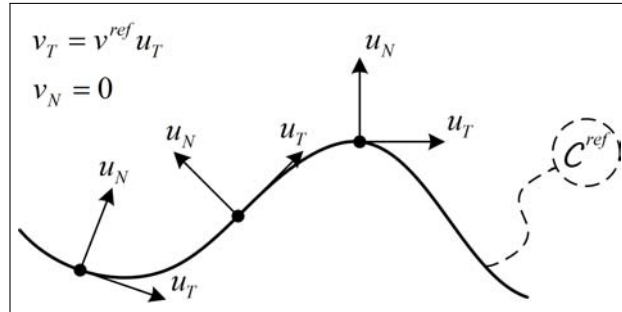


Figure 4.3: Schematic Illustration of Orthogonal Coordinate System Attached to the Reference Contour

On the other hand, the complication arise due to the mapping between the spaces where the actions exist. The reference is generated in the operational space coordinates, the error is defined in the orthogonal space coordinates and the measurement and control is carried out in the configuration space coordinates. The transformation between configuration and operational spaces is formulated in equations (27) and (28) whereas the transformation between orthogonal space and operational spaces is derived in the following subsection. In order to provide a better illustration of the mapping between spaces, a schematic representation is provided below in Figure 4.4.

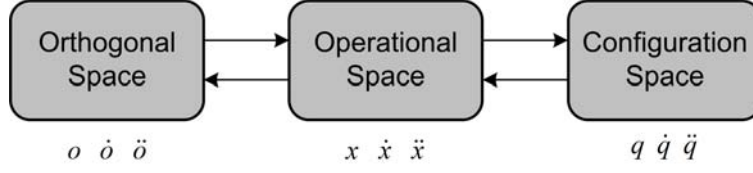


Figure 4.4: Representation of Transformations Between Spaces

4.3.1 Mapping Between Orthogonal and Operational Spaces

As mentioned earlier, the transformation between the orthogonal and operational spaces can be made using the reference curvature of contour as the tangential and normal components vary dynamically over it. Let us take the given contour reference in the operational space as;

$$\mathcal{C}^{\text{ref}}(t) = \begin{bmatrix} x^{\text{ref}}(t) \\ y^{\text{ref}}(t) \end{bmatrix} \quad (36)$$

For any time t , the unit vector \mathbf{u}_T that is tangent to the given $2D$ contour can be obtained from the normalized velocity references as follows;

$$\mathbf{u}_T = \frac{1}{\|\dot{\mathcal{C}}^{\text{ref}}(t)\|} \begin{bmatrix} \dot{x}^{\text{ref}}(t) & \dot{y}^{\text{ref}}(t) \end{bmatrix} \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} \quad (37)$$

where, $\| * \|$ stand for the L_2 norm of the corresponding identity $*$. In equation (37), \mathbf{u}_x and \mathbf{u}_y stand for the unit vectors along x and y directions respectively. Once the tangent direction is defined, the unit vector along normal direction \mathbf{u}_N can be acquired using the cross product rule $\mathbf{u}_N \times \mathbf{u}_T = 1$ as follows;

$$\mathbf{u}_N = \frac{1}{\|\dot{\mathcal{C}}^{\text{ref}}(t)\|} \begin{bmatrix} -\dot{y}^{\text{ref}}(t) & \dot{x}^{\text{ref}}(t) \end{bmatrix} \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix} \quad (38)$$

Combining the identities given in equations (37) and (38), one can formulate the transformation between the operational and orthogonal space coordinates as;

$$\mathcal{O} = \mathbf{J}_{\mathcal{O}}\mathbf{x} \quad (39)$$

where, $\mathcal{O} = [O_1(t), O_2(t)]^T$ and $\mathbf{x} = [x(t), y(t)]^T$ stand for the orthogonal space coordinates ($O_1(t)$ being the tangential component and $O_2(t)$ being the normal component) and operational space coordinates respectively. Here, the term $\mathbf{J}_{\mathcal{O}} \in \mathbb{R}^{2 \times 2}$ represent the mapping between two coordinate systems (like the forward kinematics of robotic systems) and is given by;

$$\mathbf{J}_{\mathcal{O}} = \frac{1}{\|\dot{\mathcal{C}}^{ref}(t)\|} \begin{bmatrix} \dot{x}^{ref}(t) & \dot{y}^{ref}(t) \\ -\dot{y}^{ref}(t) & \dot{x}^{ref}(t) \end{bmatrix} \quad (40)$$

Equation (39), contain the necessary transformation to switch between the positions defined in two spaces. For the derivation of controller, however, one also needs the transformation of velocities and accelerations. Differentiating (39), one can obtain;

$$\dot{\mathcal{O}} = \dot{\mathbf{J}}_{\mathcal{O}}\mathbf{x} + \mathbf{J}_{\mathcal{O}}\dot{\mathbf{x}} \quad (41)$$

$$\ddot{\mathcal{O}} = \ddot{\mathbf{J}}_{\mathcal{O}}\mathbf{x} + 2\dot{\mathbf{J}}_{\mathcal{O}}\dot{\mathbf{x}} + \mathbf{J}_{\mathcal{O}}\ddot{\mathbf{x}} \quad (42)$$

The effectiveness of smooth differentiation via closed form representation comes into picture when equation (42) is considered since the term $\ddot{\mathbf{J}}_{\mathcal{O}}$ requires the existence of third order derivatives of the reference trajectory. This is why the applicability of the controller formulated in the following subsection is limited to the curves that can be mathematically represented in closed form with smooth third order derivatives.

4.3.2 Contouring Controller

Making use of the mappings given in equations (39), (41) and (42), one can now define the tracking error as a linear combination of positions and velocities in the orthogonal space as follows;

$$\boldsymbol{\varepsilon}_{\mathcal{O}} = \mathbf{S}_{\mathcal{O}2} \left(\dot{\mathcal{O}}^{\text{ref}} - \dot{\mathcal{O}}^{\text{res}} \right) + \mathbf{S}_{\mathcal{O}1} \left(\mathcal{O}^{\text{ref}} - \mathcal{O}^{\text{res}} \right) \quad (43)$$

where, the references \mathcal{O}^{ref} , $\dot{\mathcal{O}}^{\text{ref}}$ and the responses \mathcal{O}^{res} , $\dot{\mathcal{O}}^{\text{res}}$ are obtained using the derived transformations from operational space. Here, like equation (29), $\mathbf{S}_{\mathcal{O}1} = \text{diag}\{s_1\} \in \mathbb{R}^{2 \times 2}$ and $\mathbf{S}_{\mathcal{O}2} = \text{diag}\{s_2\} \in \mathbb{R}^{2 \times 2}$ stand for the diagonal scaling matrices with positive entries (i.e. $s_1, s_2 > 0$) determining the respective weight of velocity and position over the generalized error. In equation (43) the orthogonal space velocity reference could possibly be directly taken as $\dot{\mathcal{O}}^{\text{ref}} = [v^{\text{ref}}(t), 0]^T$ where, $v^{\text{ref}}(t)$ representing the desired tangential velocity reference. However, since the transformation for orthogonal coordinate system is already derived under the approach of reference velocity for tangent direction and zero velocity for normal direction, one can directly use the transformed references from operational space. Just like the case in open contour trajectories, we can enforce exponential convergence of tracking error (43) in the orthogonal space as follows;

$$\dot{\boldsymbol{\varepsilon}}_{\mathcal{O}} + \boldsymbol{\gamma} \boldsymbol{\varepsilon}_{\mathcal{O}} = \mathbf{0} \quad (44)$$

where, $\boldsymbol{\gamma} = \text{diag}\{\gamma\} \in \mathbb{R}^{2 \times 2}$ is again diagonal matrix with $\gamma > 0$ determining the convergence rate of the error to zero. Inserting the identity from (43)

into (44), one can obtain the following dynamic equation;

$$\mathbf{S}_{\mathcal{O}2} \left(\ddot{\mathcal{O}}^{\text{ref}} - \ddot{\mathcal{O}}^{\text{res}} \right) + (\mathbf{S}_{\mathcal{O}1} + \gamma \mathbf{S}_{\mathcal{O}2}) \left(\dot{\mathcal{O}}^{\text{ref}} - \dot{\mathcal{O}}^{\text{res}} \right) + \gamma \mathbf{S}_{\mathcal{O}1} (\mathcal{O}^{\text{ref}} - \mathcal{O}^{\text{res}}) = \mathbf{0} \quad (45)$$

which can be recast as;

$$\left(\ddot{\mathcal{O}}^{\text{ref}} - \ddot{\mathcal{O}}^{\text{res}} \right) + \mathbf{K}_{\mathcal{O}2} \left(\dot{\mathcal{O}}^{\text{ref}} - \dot{\mathcal{O}}^{\text{res}} \right) + \mathbf{K}_{\mathcal{O}1} (\mathcal{O}^{\text{ref}} - \mathcal{O}^{\text{res}}) = \mathbf{0} \quad (46)$$

where, $\mathbf{K}_{\mathcal{O}2} \in \mathbb{R}^{2 \times 2}$ and $\mathbf{K}_{\mathcal{O}1} \in \mathbb{R}^{2 \times 2}$ respectively stand for the derivative and proportional gains of the system to enforce convergence of contouring error to zero value and have the following content;

$$\mathbf{K}_{\mathcal{O}2} = \mathbf{S}_{\mathcal{O}2}^{-1} (\mathbf{S}_{\mathcal{O}1} + \gamma \mathbf{S}_{\mathcal{O}2}) \quad (47)$$

$$\mathbf{K}_{\mathcal{O}1} = \mathbf{S}_{\mathcal{O}2}^{-1} \gamma \mathbf{S}_{\mathcal{O}1} \quad (48)$$

Since the control action is handled in the configuration space, the dynamics given in equation (46) has to be transformed first to operational space and then to configuration space. In order to decrease the complexity in the equations, let us first define the following error which can also be perceived as an intermediate coordinate system;

$$\mathbf{e}_x = \mathbf{x}^{\text{ref}} - \mathbf{x}^{\text{res}} \quad (49)$$

Here, one has to recall the existence of 3^{rd} dimension in operational space for a generalized fully actuated 3-DOF robotic system. Hence, in equation (49), the operational space error is defined as a 3D vector (i.e. $\mathbf{e}_x \in \mathbb{R}^{3 \times 1}$) assuming some constant reference value for z-axis. Now, in light of transformations

given in (39), (41) and (42), one can transform the dynamics given in equation (46) to operational space coordinates as;

$$\left(\ddot{\mathbf{J}}_{\mathcal{O}}\mathbf{e}_x + 2\dot{\mathbf{J}}_{\mathcal{O}}\dot{\mathbf{e}}_x + \mathbf{J}_{\mathcal{O}}\ddot{\mathbf{e}}_x\right) + \mathbf{K}_{\mathcal{O}2}\left(\dot{\mathbf{J}}_{\mathcal{O}}\mathbf{e}_x + \mathbf{J}_{\mathcal{O}}\dot{\mathbf{e}}_x\right) + \mathbf{K}_{\mathcal{O}1}\mathbf{J}_{\mathcal{O}}\mathbf{e}_x = \mathbf{0} \quad (50)$$

which can be rearranged as;

$$\ddot{\mathbf{e}}_x + \left(2\mathbf{J}_{\mathcal{O}}^{-1}\dot{\mathbf{J}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}}^{-1}\mathbf{K}_{\mathcal{O}2}\mathbf{J}_{\mathcal{O}}\right)\dot{\mathbf{e}}_x + \left(\mathbf{J}_{\mathcal{O}}^{-1}\ddot{\mathbf{J}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}}^{-1}\mathbf{K}_{\mathcal{O}2}\dot{\mathbf{J}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}}^{-1}\mathbf{K}_{\mathcal{O}1}\mathbf{J}_{\mathcal{O}}\right)\mathbf{e}_x = \mathbf{0} \quad (51)$$

Equation (51), represents the operational space dynamics of system that can track the desired references in the orthogonal space coordinates. In order to generate the necessary control input for the joints, however, this dynamics should be converted from operational space coordinates to configuration space coordinates. Similar to the operational space case, one can define the following error for the configuration space references;

$$\mathbf{e}_q = \mathbf{q}^{\text{ref}} - \mathbf{q}^{\text{res}} \quad (52)$$

Benefitting from the linearity, we can transform between \mathbf{e}_x and \mathbf{e}_q using the transformations given in (27) and (28). Hence, one can convert equation (51) to the configuration space coordinates as follows;

$$\begin{aligned} &\mathbf{J}_x\ddot{\mathbf{e}}_q + \left(\dot{\mathbf{J}}_x + \left(2\mathbf{J}_{\mathcal{O}}^{-1}\dot{\mathbf{J}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}}^{-1}\mathbf{K}_{\mathcal{O}2}\mathbf{J}_{\mathcal{O}}\right)\mathbf{J}_x\right)\dot{\mathbf{e}}_q \\ &+ \left(\mathbf{J}_{\mathcal{O}}^{-1}\ddot{\mathbf{J}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}}^{-1}\mathbf{K}_{\mathcal{O}2}\dot{\mathbf{J}}_{\mathcal{O}} + \mathbf{J}_{\mathcal{O}}^{-1}\mathbf{K}_{\mathcal{O}1}\mathbf{J}_{\mathcal{O}}\right)\mathcal{F}(\mathbf{e}_q) = \mathbf{0} \end{aligned} \quad (53)$$

Now, we can rearrange equation (53) and substitute back the error content from (52) to obtain the dynamics in the configuration space as;

$$\ddot{\mathbf{q}}^{\text{ref}} - \ddot{\mathbf{q}}^{\text{res}} + \mathbf{K}_v (\dot{\mathbf{q}}^{\text{ref}} - \dot{\mathbf{q}}^{\text{res}}) + \mathbf{K}_p (\mathcal{F}(\mathbf{q}^{\text{ref}}) - \mathcal{F}(\mathbf{q}^{\text{res}})) = \mathbf{0} \quad (54)$$

Here, the relative gains of position error $\mathbf{K}_p \in \mathbb{R}^{3 \times 3}$ and velocity error $\mathbf{K}_v \in \mathbb{R}^{3 \times 3}$ (i.e. the proportional and derivative gains) can be given as;

$$\mathbf{K}_v = \mathbf{J}_x^{-1} \dot{\mathbf{J}}_x + 2\mathbf{J}_x^{-1} \mathbf{J}_o^{-1} \dot{\mathbf{J}}_o \mathbf{J}_x + \mathbf{J}_x^{-1} \mathbf{J}_o^{-1} \mathbf{K}_{o2} \mathbf{J}_o \mathbf{J}_x \quad (55)$$

$$\mathbf{K}_p = \mathbf{J}_x^{-1} \mathbf{J}_o^{-1} \ddot{\mathbf{J}}_o + \mathbf{J}_x^{-1} \mathbf{J}_o^{-1} \mathbf{K}_{o2} \dot{\mathbf{J}}_o + \mathbf{J}_x^{-1} \mathbf{J}_o^{-1} \mathbf{K}_{o1} \mathbf{J}_o \quad (56)$$

Under the condition that the plant under consideration is disturbance compensated with the structure shown in Figure 4.1, the desired acceleration that will satisfy the tracking of desired tangential velocity reference and zero normal velocity reference can be given as;

$$\ddot{\mathbf{q}}^{\text{des}} = \ddot{\mathbf{q}}^{\text{ref}} + \mathbf{K}_v (\dot{\mathbf{q}}^{\text{ref}} - \dot{\mathbf{q}}^{\text{res}}) + \mathbf{K}_p (\mathcal{F}(\mathbf{q}^{\text{ref}}) - \mathcal{F}(\mathbf{q}^{\text{res}})) \quad (57)$$

which concludes the derivation of contouring controller.

Elaborating over the content of the contouring controller, one can observe the similarity in the structure to that of independent joint control. However, there is one major point that makes this control scheme superior to the standard independent joint control structure. The advantage of contouring controller lies in the ability of directly acting over the error defined in the orthogonal space. This way, the acceleration induced forces that exist in independent joint control is taken into consideration and fed forward to the system via the coefficients \mathbf{K}_p and \mathbf{K}_v . The results acquired from the

contouring control structure will be presented in Chapter 8 while a schematic diagram of the controller structure is given below in Figure 4.5.

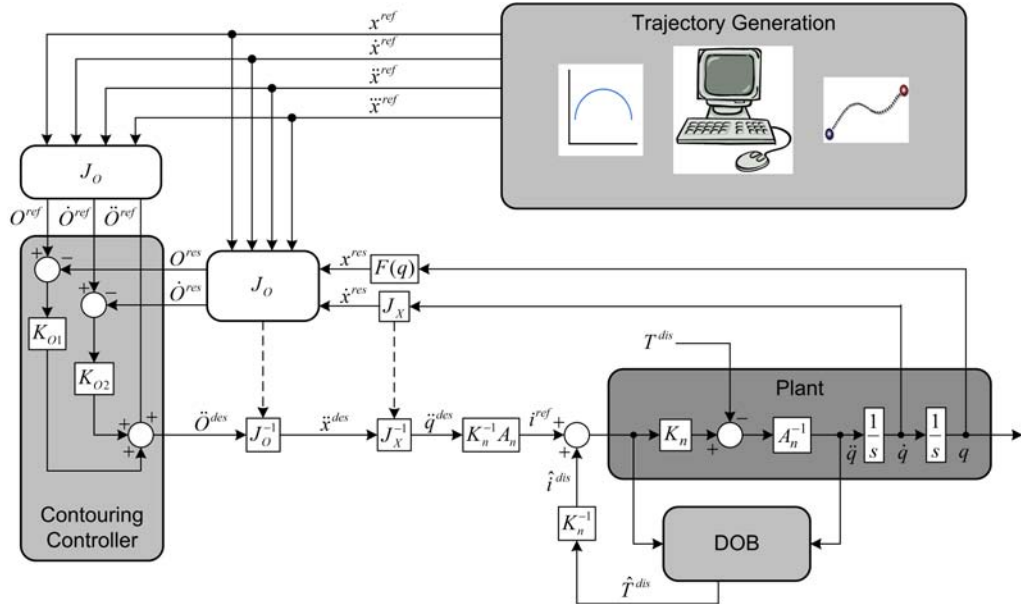


Figure 4.5: Structure of the Contouring Controller

Chapter V

5 Operator Assisted Control for Non-Contact Machining

Fully automated non-contact machining techniques presented in the previous chapter provide an environment for cutting arbitrary shapes with certain desired depth. The depth of cutting is incorporated in the tangential velocity reference while the controller's main task is to preserve that velocity reference during operation. However, in certain applications, the desired cutting depth of the given shape is unknown prior to process. In such cases, systems with the requirement of manual machining capability gains more importance. Semi-automated or manual machining techniques have been more frequently used in medical applications [69]. Among the medical applications of robotic non-contact machining, researchers put particular attention on robot assisted laser surgery systems [70], [71]. Successful implementations on orthopedic surgery [72], laser scalping [73] and bone cutting [74] have been illustrated in the literature by some authors.

Looking from the motion control point of view, the robotic system is required to function in such a way that a human operator can follow arbitrary reference shape of desired contour. In other words, task of the motion control system is constraining the operator motion on a certain pre-defined trajec-

tory. Especially in laser surgery systems, any undesired motion beyond the constraining contour would lead to degenerations in the success rates of the operation due to undesired cutting of healthy tissue. Hence, in order to assist the surgeon during operation, certain virtual constraints are required to be satisfied by the motion control system.

Realization of motion constraining controller can be possible either directly on the tool over which the laser system is attached [72] or in a bilateral system framework [114]. In the bilateral control framework, the overall robotic system is decomposed into two constituents referred to as the master and slave systems respectively. The master system (manually controlled by the operator) is assigned the task of constraining the operator's motion to the desired contour, while the slave system (coupled with the laser unit) is desired to track the motion executed on the master system. This way, the reference contour can be traced by the operator and the cutting operation can be carried out by the slave system based on the manual position reference enforced on the master system. An illustration of this structure is given below in Figure 5.1.

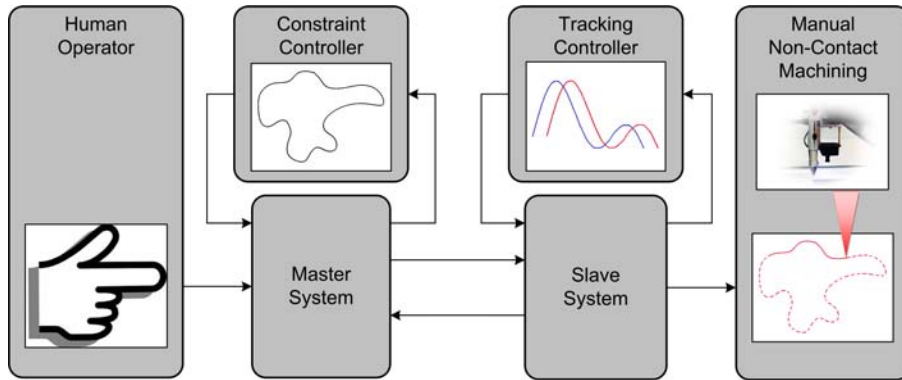


Figure 5.1: Representative Structure of Non-Contact Assistive Machining

In the derivation given below, an assistive manual non-contact machining

scheme is presented for application on industrial and/or surgical systems requiring remote operation. The structure is realized on the basis of constraint generation from an external image and implemented on a master-slave motion tracking platform. In that sense, the trajectory data acquired from an image is used as the constraint and the master system controller is formulated to satisfy the desired motion constraint. Furthermore, secondary controllers are formulated to enforce the slave system track the motion exhibited by a human operator on the master system for cases with and without time delay. This way, the operator's motion is confined to be free within certain boundaries on the master side while the slave system does the machining following the motion exhibited on the master side.

5.1 Constraints in Operational Space

As core part of this study, the system under consideration is required to enforce motion under certain constraints which are defined as functions of operational space coordinates. Let us assume that the constraints defined on an n -*DOF* system is represented by the vector valued continuous function of operational space coordinates as;

$$\phi(\mathbf{x}) = \mathbf{0} \tag{58}$$

where, we have $\phi \in \mathbb{R}^{m \times 1}$ with $m < n$. The functional relationship between operational space coordinates and the corresponding constraints defined by equation (58) is similar to the mapping between configuration space and operational space as defined in the preceding chapters. In that sense, one can also define the relationship between the time rate of changes of constraints

and operational space coordinates in a similar way as follows;

$$\dot{\phi} = \mathbf{J}_\phi \dot{\mathbf{x}} \quad (59)$$

$$\ddot{\phi} = \mathbf{J}_\phi \ddot{\mathbf{x}} + \dot{\mathbf{J}}_\phi \dot{\mathbf{x}} \quad (60)$$

where, $\mathbf{J}_\phi = \mathbf{J}_\phi(\mathbf{x}) = (\partial\phi(\mathbf{x})/\partial\mathbf{x})$ is the jacobian matrix associated with the constraints. The constraint jacobian $\mathbf{J}_\phi \in \mathbb{R}^{m \times n}$ is assumed to have full row rank.

Enforcement of the constraints defined over the operational space coordinates can be possible by a controller integrated to the actuators in configuration space. In the following subsection, the derivation of controller to satisfy the requirements desired by the constraints is made to obtain the configuration space acceleration references.

5.2 Constraint Controller Derivation

In order to formulate the structure of controller, one has to define the error in the constrained direction. Let us select the structure of error as a linear combination of the constraint and its derivative as follows;

$$\boldsymbol{\varepsilon}_\phi = \mathbf{C}_2 \left(\dot{\phi}^{\text{ref}} - \dot{\phi}^{\text{res}} \right) + \mathbf{C}_1 \left(\phi^{\text{ref}} - \phi^{\text{res}} \right) \quad (61)$$

where, $\boldsymbol{\varepsilon}_\phi \in \mathbb{R}^{m \times 1}$, $\mathbf{C}_1 = \text{diag}\{c_1\} \in \mathbb{R}^{m \times m}$ and $\mathbf{C}_2 = \text{diag}\{c_2\} \in \mathbb{R}^{m \times m}$ stand for the constraint tracking error and positive definite (i.e. $c_1, c_2 > 0$) diagonal matrices of weighting factors for constraint error and its rate of change respectively. Unlike the structure of tracking errors used in the previous chapters, in equation (61), the reference constraint and its time

rate of change is by definition set as zero (i.e. $\phi^{\text{ref}} = \mathbf{0}$ and $\dot{\phi}^{\text{ref}} = \mathbf{0}$). Hence, for simplicity of the analysis in the rest of this chapter, the terms ϕ^{ref} and $\dot{\phi}^{\text{ref}}$ are omitted. Furthermore, since the objective of controller is to push the tracking error to zero value, one can alternate the remaining minus signs of equation (61). This way, one can reformulate the structure of constraint tracking error as;

$$\boldsymbol{\varepsilon}_\phi = \mathbf{C}_2 \dot{\phi}^{\text{res}} + \mathbf{C}_1 \phi^{\text{res}} \quad (62)$$

Like the previous cases, enforcement of exponential convergence for the operational space constraints to zero value can be possible by the enforcement of following dynamics;

$$\dot{\boldsymbol{\varepsilon}}_\phi + \mathbf{K} \boldsymbol{\varepsilon}_\phi = \mathbf{0} \quad (63)$$

with $\mathbf{K} \in \mathbb{R}^{m \times m}$ being the positive definite diagonal matrix whose entries k_{ii} for $i = 1, \dots, m$ determine the exponential convergence rate of error for the i^{th} constraint. For a system with decoupled constraints, the matrix \mathbf{K} will have a diagonal structure (i.e. $k_{ij} = 0$ for $i \neq j$). Proceeding further for the controller derivation, one can substitute equation (62) into (63) obtaining the following dynamics for the constraint;

$$\mathbf{C}_2 \ddot{\phi}^{\text{res}} + (\mathbf{C}_1 + \mathbf{K} \mathbf{C}_2) \dot{\phi}^{\text{res}} + (\mathbf{K} \mathbf{C}_1) \phi^{\text{res}} = \mathbf{0} \quad (64)$$

which can be rearranged to give the following dynamics in the acceleration dimension;

$$\ddot{\phi}^{\text{res}} + \mathbf{K}_{\phi_2} \dot{\phi}^{\text{res}} + \mathbf{K}_{\phi_1} \phi^{\text{res}} = \mathbf{0} \quad (65)$$

where, the content of gains are given as;

$$\mathbf{K}_{\phi 2} = \mathbf{C}_2^{-1} (\mathbf{C}_1 + \mathbf{K}\mathbf{C}_2) \quad (66)$$

$$\mathbf{K}_{\phi 1} = \mathbf{C}_2^{-1} \mathbf{K}\mathbf{C}_1 \quad (67)$$

The dynamics in the acceleration dimension defined by equation (65) provides the system with exponential decay of error to zero as long as proper selection of convergence gain \mathbf{K} is made. In order to enforce this dynamics in constraint space, one has to find out the value of corresponding desired acceleration in configuration space. This comes out of two consecutive transformation between constraint, operational and configuration space coordinates.

Let us find the desired dynamics in operational space first. To do so, one can benefit from the identities given in equations (59) and (60) and substitute into equation (65) to obtain;

$$\mathbf{J}_\phi \ddot{\mathbf{x}}^{\text{res}} + \left(\dot{\mathbf{J}}_\phi + \mathbf{K}_{\phi 2} \mathbf{J}_\phi \right) \dot{\mathbf{x}}^{\text{res}} + \mathbf{K}_{\phi 1} \phi(\mathbf{x}^{\text{res}}) = \mathbf{0} \quad (68)$$

which can be recast as;

$$\ddot{\mathbf{x}}^{\text{res}} + \mathbf{K}_{\mathbf{x}2} \dot{\mathbf{x}}^{\text{res}} + \mathbf{K}_{\mathbf{x}1} \phi(\mathbf{x}^{\text{res}}) = \mathbf{0} \quad (69)$$

Here, the coefficients standing for the proportional and derivative gains can be given as;

$$\mathbf{K}_{\mathbf{x}1} = \mathbf{J}_\phi^\# \mathbf{K}_{\phi 1} \quad (70)$$

$$\mathbf{K}_{\mathbf{x}2} = \mathbf{J}_\phi^\# \left(\dot{\mathbf{J}}_\phi + \mathbf{K}_{\phi 2} \mathbf{J}_\phi \right) \quad (71)$$

where, the term $\mathbf{J}_\phi^\# \in \mathbb{R}^{n \times m}$ stand for the right weighted pseudo-inverse of

the matrix \mathbf{J}_ϕ . Equation (69) represent the dynamics in operational space to enforce the exponentially decaying error condition for the satisfaction of constraint $\phi(\mathbf{x})$. Particularly, the content of coefficient \mathbf{K}_{x2} represent the coupling of velocities in the constrained dimension necessary for precise tracking requirement. As the controller is acting in the configuration space coordinates, in order to obtain the necessary input for an acceleration controllable plant, one has make a final transformation from operational space to configuration space coordinates. This can be accomplished via the transformations given in (25), (27) and (28). Substituting back from these identities to equation (69), one can acquire;

$$\mathbf{J}_x \ddot{\mathbf{q}}^{\text{res}} + \left(\dot{\mathbf{J}}_x + \mathbf{K}_{x2} \mathbf{J}_x \right) \dot{\mathbf{q}}^{\text{res}} + \mathbf{K}_{x1} \phi(\mathcal{F}(\mathbf{q}^{\text{res}})) = \mathbf{0} \quad (72)$$

which can be further simplified as;

$$\ddot{\mathbf{q}}^{\text{res}} + \mathbf{K}_{q2} \dot{\mathbf{q}}^{\text{res}} + \mathbf{K}_{q1} \phi(\mathcal{F}(\mathbf{q}^{\text{res}})) = \mathbf{0} \quad (73)$$

where, the gains \mathbf{K}_{q1} and \mathbf{K}_{q2} have the following content;

$$\mathbf{K}_{q1} = \mathbf{J}_x^{-1} \mathbf{K}_{x1} \quad (74)$$

$$\mathbf{K}_{q2} = \mathbf{J}_x^{-1} \left(\dot{\mathbf{J}}_x + \mathbf{K}_{x2} \mathbf{J}_x \right) \quad (75)$$

Making necessary substitutions from equations (66), (67), (70) and (71), one can write down the full content of these gains as;

$$\mathbf{K}_{q1} = \mathbf{J}_x^{-1} \mathbf{J}_\phi^\# \mathbf{C}_2^{-1} \mathbf{K} \mathbf{C}_1 \quad (76)$$

$$\mathbf{K}_{\mathbf{q}2} = \mathbf{J}_{\mathbf{x}}^{-1} \dot{\mathbf{J}}_{\mathbf{x}} + \mathbf{J}_{\mathbf{x}}^{-1} \mathbf{J}_{\phi}^{\#} \dot{\mathbf{J}}_{\phi} \mathbf{J}_{\mathbf{x}} + \mathbf{J}_{\mathbf{x}}^{-1} \mathbf{J}_{\phi}^{\#} \mathbf{C}_2^{-1} \mathbf{C}_1 \mathbf{J}_{\phi} \mathbf{J}_{\mathbf{x}} + \mathbf{J}_{\mathbf{x}}^{-1} \mathbf{J}_{\phi}^{\#} \mathbf{C}_2^{-1} \mathbf{K} \mathbf{C}_2 \mathbf{J}_{\phi} \mathbf{J}_{\mathbf{x}} \quad (77)$$

Equation (73), reflects the dynamics of constraint tracking in configuration space coordinates with coefficients (76) and (77). The desired acceleration can be retrieved from this equation as follows;

$$\ddot{\mathbf{q}}^{\text{des}} = -\mathbf{K}_{\mathbf{q}2} \dot{\mathbf{q}}^{\text{res}} - \mathbf{K}_{\mathbf{q}1} \phi(\mathcal{F}(\mathbf{q}^{\text{res}})) \quad (78)$$

Under the condition that satisfactory disturbance rejection exists, the desired acceleration given in (78) will enforce exponential convergence of motion to the desired constraint. This completes the discussion regarding the structure of constraint controller. A block diagram of the controller is provided below in Figure 5.2 for better illustration of the process. In Appendix A, details related to the realization of this controller to provide free spatial motion along a pre-defined curve in task space is explained.

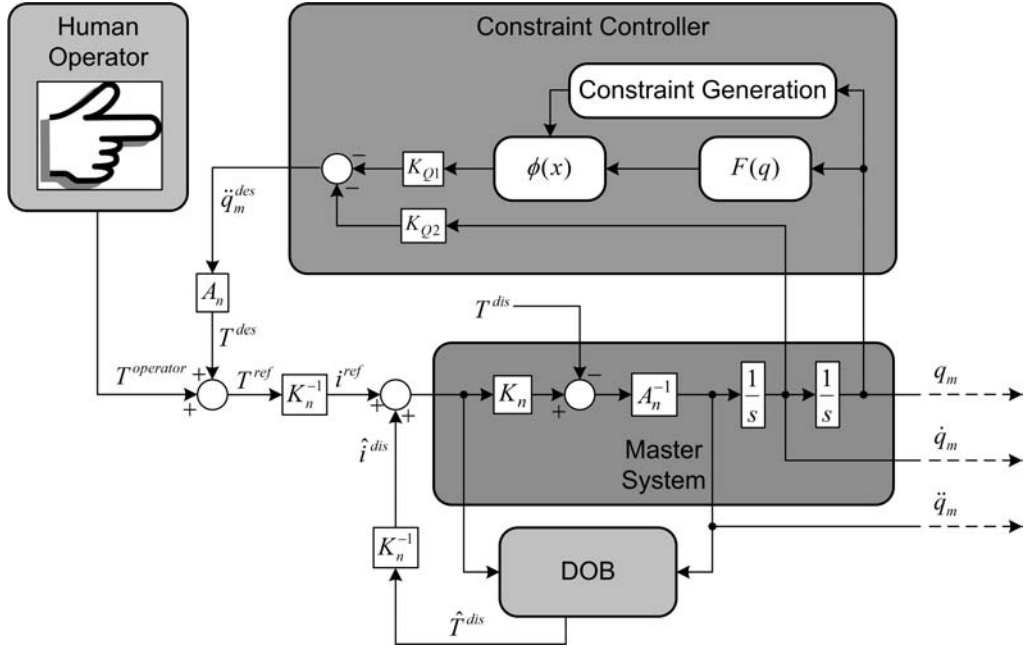


Figure 5.2: Structure of the Master Side Motion Constraining Controller

5.3 Slave System Tracking Controller

In order to transfer the motion from master system to the slave system, we need to formulate another controller. The objective of this controller is to synchronize the motion between the two systems. This synchronization can be realized for systems with and without communication time delay utilizing different controller structures respectively. Under the assumption that disturbances acting over the slave system are cleared out using the formulation described in equation (23), the slave plant can assume acceleration references. In that sense, the procedure to derive the controller for motion synchronization should end up with acquisition of desired accelerations. Since the content being under consideration does not include feedback of forces from slave to master system (i.e. master system is allowed to have free motion in a con-

strained workspace), these are the desired accelerations of slave plant prior to the motion reference acquired from master system.

5.3.1 Controller Without Time-Delay

For indoor applications that do not require transmission of signals to distant locations, one can assume negligibly small amount of time delay between master and slave systems. Here, the controller will be derived based on this assumption.

We can start by defining a tracking error between master and slave systems. Taking into consideration that there may exist possible scaling between motions of master and slave systems, one can define a linear combination of position and velocity errors as follows;

$$\boldsymbol{\varepsilon}_{\text{ms}} = \mathbf{W}_2 (\dot{\mathbf{x}}_{\text{m}} - \boldsymbol{\beta}\dot{\mathbf{x}}_{\text{s}}) + \mathbf{W}_1 (\mathbf{x}_{\text{m}} - \boldsymbol{\beta}\mathbf{x}_{\text{s}}) \quad (79)$$

where, $\mathbf{W}_2 \in \mathbb{R}^{n \times n}$, $\mathbf{W}_1 \in \mathbb{R}^{n \times n}$ and $\boldsymbol{\beta} \in \mathbb{R}^{n \times n}$ are diagonal matrices standing for the weight of velocities in tracking error, weight of positions in tracking error and motion scaling between master and slave robots. In order to prevent further couplings on the controller and preserve the aspect ratio of motion between two systems, these three matrices can be assumed as scalar multiples of $n \times n$ identity matrix (i.e. $\mathbf{W}_2 = w_2 \mathbf{I}_n$, $\mathbf{W}_1 = w_1 \mathbf{I}_n$ and $\boldsymbol{\beta} = \beta \mathbf{I}_n$). Similar to the constraint controller, the tracking error (79) will converge to zero value if it is enforced to have an exponentially decaying dynamics. Mathematically speaking, this requires the realization of the following equation;

$$\dot{\boldsymbol{\varepsilon}}_{\text{ms}} + \mathbf{K}_{\text{ms}} \boldsymbol{\varepsilon}_{\text{ms}} = \mathbf{0} \quad (80)$$

where, $\mathbf{K}_{\text{ms}} \in \mathbb{R}^{n \times n}$ is the diagonal positive definite matrix that determines the convergence rate of tracking error to zero. Substituting equation (79) into (80) yields the following expression;

$$\mathbf{W}_2 (\ddot{\mathbf{x}}_{\text{m}} - \beta \ddot{\mathbf{x}}_{\text{s}}) + (\mathbf{W}_1 + \mathbf{K}_{\text{ms}} \mathbf{W}_2) (\dot{\mathbf{x}}_{\text{m}} - \beta \dot{\mathbf{x}}_{\text{s}}) + \mathbf{K}_{\text{ms}} \mathbf{W}_1 (\mathbf{x}_{\text{m}} - \beta \mathbf{x}_{\text{s}}) = \mathbf{0} \quad (81)$$

Equation (81) enforces the error between master and slave systems to zero value by proper selection of convergence gain \mathbf{K}_{ms} . The tracking requirement of slave system would be satisfied with the following operational space reference acceleration;

$$\ddot{\mathbf{x}}_{\text{s}} = \Lambda_1 \ddot{\mathbf{x}}_{\text{m}} + \Lambda_2 (\dot{\mathbf{x}}_{\text{m}} - \beta \dot{\mathbf{x}}_{\text{s}}) + \Lambda_3 (\mathbf{x}_{\text{m}} - \beta \mathbf{x}_{\text{s}}) \quad (82)$$

where, the coefficients have the following content;

$$\Lambda_1 = \beta^{-1} \quad (83)$$

$$\Lambda_2 = \beta^{-1} \mathbf{W}_2^{-1} \mathbf{W}_1 + \beta^{-1} \mathbf{W}_2^{-1} \mathbf{K}_{\text{ms}} \mathbf{W}_2 \quad (84)$$

$$\Lambda_3 = \beta^{-1} \mathbf{W}_2^{-1} \mathbf{K}_{\text{ms}} \mathbf{W}_1 \quad (85)$$

For system under consideration, since both measurement and actuation are carried out in the configuration space, one can make use of the identities given in (27) and (28) to convert equation (82) into an expression that only includes configuration space coordinates as follows;

$$\begin{aligned} \mathbf{J}_{\text{s}} \ddot{\mathbf{q}}_{\text{s}} + \dot{\mathbf{J}}_{\text{s}} \dot{\mathbf{q}}_{\text{s}} &= \Lambda_1 \mathbf{J}_{\text{m}} \ddot{\mathbf{q}}_{\text{m}} + \Lambda_1 \dot{\mathbf{J}}_{\text{m}} \dot{\mathbf{q}}_{\text{m}} + \Lambda_2 \mathbf{J}_{\text{m}} \dot{\mathbf{q}}_{\text{m}} - \Lambda_2 \beta \mathbf{J}_{\text{s}} \dot{\mathbf{q}}_{\text{s}} \\ &+ \Lambda_3 \mathcal{F}(\mathbf{q}_{\text{m}}) - \Lambda_3 \beta \mathcal{F}(\mathbf{q}_{\text{s}}) \end{aligned} \quad (86)$$

As pointed out earlier, enforcement of tracking can be possible by feeding the desired acceleration calculated for the slave system. Hence, one can rearrange equation (86) and get the slave plant desired acceleration as follows;

$$\ddot{\mathbf{q}}_s^{\text{des}} = \Psi_1 \ddot{\mathbf{q}}_m + \Psi_2 \dot{\mathbf{q}}_m - \Psi_3 \dot{\mathbf{q}}_s + \Psi_4 \mathcal{F}(\mathbf{q}_m) - \Psi_5 \mathcal{F}(\mathbf{q}_s) \quad (87)$$

where, the terms $\Psi_k \in \mathbb{R}^{n \times n}$ for $k = 1, 2, \dots, 5$ have the following contents;

$$\Psi_1 = \mathbf{J}_s^{-1} \boldsymbol{\beta}^{-1} \mathbf{J}_m \quad (88)$$

$$\Psi_2 = \mathbf{J}_s^{-1} \boldsymbol{\beta}^{-1} \left(\dot{\mathbf{J}}_m + \mathbf{W}_2^{-1} \mathbf{W}_1 \mathbf{J}_m + \mathbf{W}_2^{-1} \mathbf{K}_{ms} \mathbf{W}_2 \mathbf{J}_m \right) \quad (89)$$

$$\Psi_3 = \mathbf{J}_s^{-1} \boldsymbol{\beta}^{-1} \left(\boldsymbol{\beta} \dot{\mathbf{J}}_s + \mathbf{W}_2^{-1} \mathbf{W}_1 \boldsymbol{\beta} \mathbf{J}_s + \mathbf{W}_2^{-1} \mathbf{K}_{ms} \mathbf{W}_2 \boldsymbol{\beta} \mathbf{J}_s \right) \quad (90)$$

$$\Psi_4 = \mathbf{J}_s^{-1} \boldsymbol{\beta}^{-1} \mathbf{W}_2^{-1} \mathbf{K}_{ms} \mathbf{W}_1 \quad (91)$$

$$\Psi_5 = \mathbf{J}_s^{-1} \boldsymbol{\beta}^{-1} \mathbf{W}_2^{-1} \mathbf{K}_{ms} \mathbf{W}_1 \boldsymbol{\beta} \quad (92)$$

The desired acceleration (87) derived for the slave plant will enforce tracking of master motion as long as the plant is in acceleration controllable form. A depiction of the slave system tracking controller block diagram is shown in Figure 5.3 below.

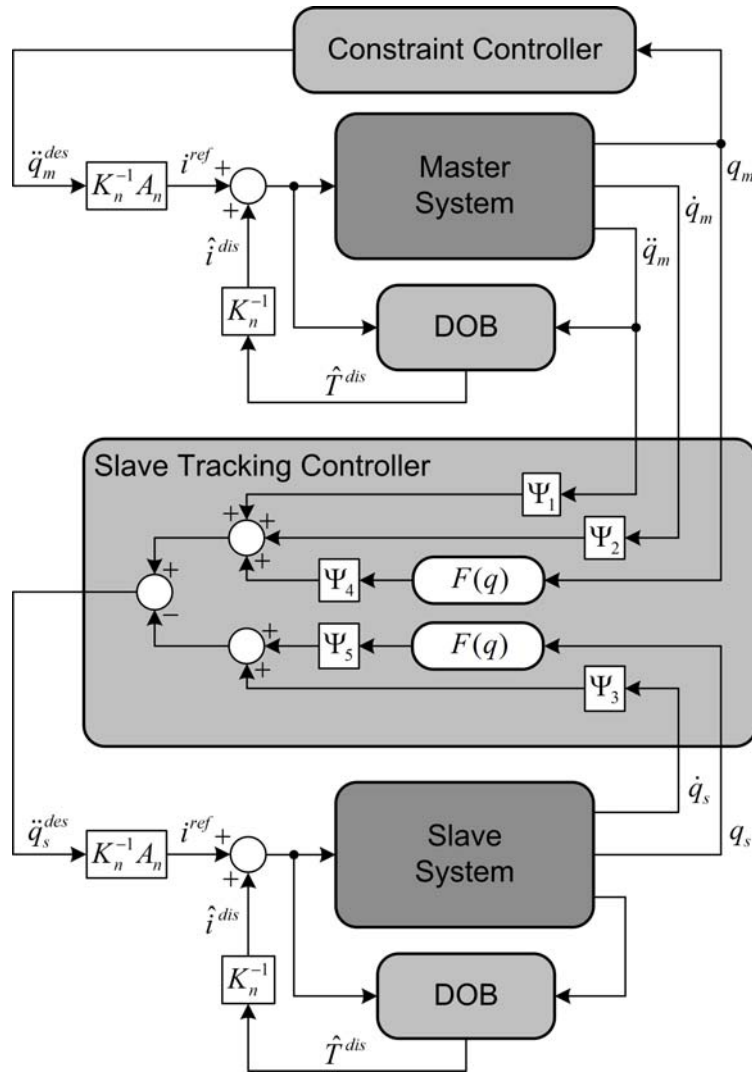


Figure 5.3: Structure of the Slave System Tracking Controller Without Time Delay

5.3.2 Controller With Time-Delay

For applications involving machining operation from a remote distance, usage of network medium is a mandatory situation. However, network environment

is sensitive to time delay in data transmission. Due to the nature of delay, both control input and measurements from the remote plant will be nonlinearly distorted. In order to have stable tracking of master position reference by the slave system under time delay, it is necessary to have an observer that can overcome the effect of measurement delay from the slave side. As long as matching conditions are satisfied and slave motion is estimated correctly, the same master side controller (87) can be used to create the necessary control input for position tracking. The following analysis includes the implementation of a previously proposed network observer structure [115] for a multi-DOF system. It should be noted here that the measurement from all slave system joints are transmitted in the same data package, hence encountering the same amount of delay.

Assuming position and velocity of the slave plant is available at the remote location, when there is delay D in the measurement channel, controller can only access the delayed position and velocity (i.e. $\mathbf{q}_s(t - D)$ and $\dot{\mathbf{q}}_s(t - D)$). So, it is necessary to use an observer that can give the predictions of the remote plant position and velocity synchronized with the actual outputs, that is without any delays. In that sense, one can define that the goal is to design observer based on available measurements $\mathbf{q}_s(t - D)$, $\dot{\mathbf{q}}_s(t - D)$ and control input $\ddot{\mathbf{q}}_s^{\text{des}}$ with zero estimation error. Once the actual output of the slave plant is estimated, then this can be used in another controller to generate the input to the remote plant.

Under the assumption that slave plant has nominal behavior with integrated DOB and perceiving the total effect of measurement delay as a network disturbance acting on the estimator, one can introduce a second observer over the network to come up with an initial estimate of the slave plant velocity.

Using the available data, the total network disturbance can be given as follows

$$\mathbf{T}_{nw}^{\text{dis}}(t) = \ddot{\mathbf{q}}_s^{\text{des}}(t) - \mathbf{A}_n \ddot{\mathbf{q}}_s(t - D) \quad (93)$$

Just like the conventional disturbance observer structure, the estimation of this network disturbance can be made using delayed slave plant velocity and a low pass filter. The estimated network disturbance stand for the torque that is supposed to act on the slave plant during measurement delay. Since the slave plant is enforced to behave nominal with DOB, the estimated network disturbance can be passed through the nominal inertia of slave plant and be integrated to give the velocity difference that is supposed to exist during the delay time. In mathematical terms, this can be expressed as

$$\Delta \dot{\mathbf{q}}_s(t) = \int \mathbf{A}_n^{-1} \mathbf{T}_{nw}^{\text{dis}}(\xi) d\xi \quad (94)$$

Addition of this velocity difference to the delayed velocity gives the estimated velocity of the slave plant as shown below:

$$\hat{\mathbf{q}}_s(t) = \mathbf{q}_s(t - D) + \Delta \dot{\mathbf{q}}_s(t) \quad (95)$$

Under the assumption that initial conditions of the slave system and the observer output are the same and perfect disturbance cancelation is available, this observer can track the slave velocity without any delay. However, there is no guarantee about the initial conditions being equal and perfect disturbance rejection is not always feasible. So, one has to insert additional compensation to push the estimated states (i.e. position and velocity) to the actual slave plant output. The details related to this additional compensation, named as

the convergence term, is analyzed in [115] and is beyond the context of this study. It is important to note here that the precise convergence of estimated slave states to the actual states under the existence of measurement and control channels' time delays have been proven in that study.

Originating from this point, the estimated slave position ($\hat{\mathbf{q}}_s$) and velocity ($\hat{\dot{\mathbf{q}}}_s$) can be used on the master side controller derived in the previous section. This way, the reference acceleration for the slave system when there is time delay gets the form;

$$\ddot{\mathbf{q}}_s^{\text{des}} = \Psi_1 \ddot{\mathbf{q}}_m + \Psi_2 \dot{\mathbf{q}}_m - \Psi_3 \hat{\dot{\mathbf{q}}}_s + \Psi_4 \mathcal{F}(\mathbf{q}_m) - \Psi_5 \mathcal{F}(\hat{\mathbf{q}}_s) \quad (96)$$

with the same gains given in equations (88) to (92). One main difference of this controller structure is that the slave system receives the desired acceleration after the control channel data transmission and hence tracks the reference with certain amount of time delay. In order to complete the analysis, a block diagram illustration of the whole slave system controller and observer for time delayed case is given below in Figure 5.4.

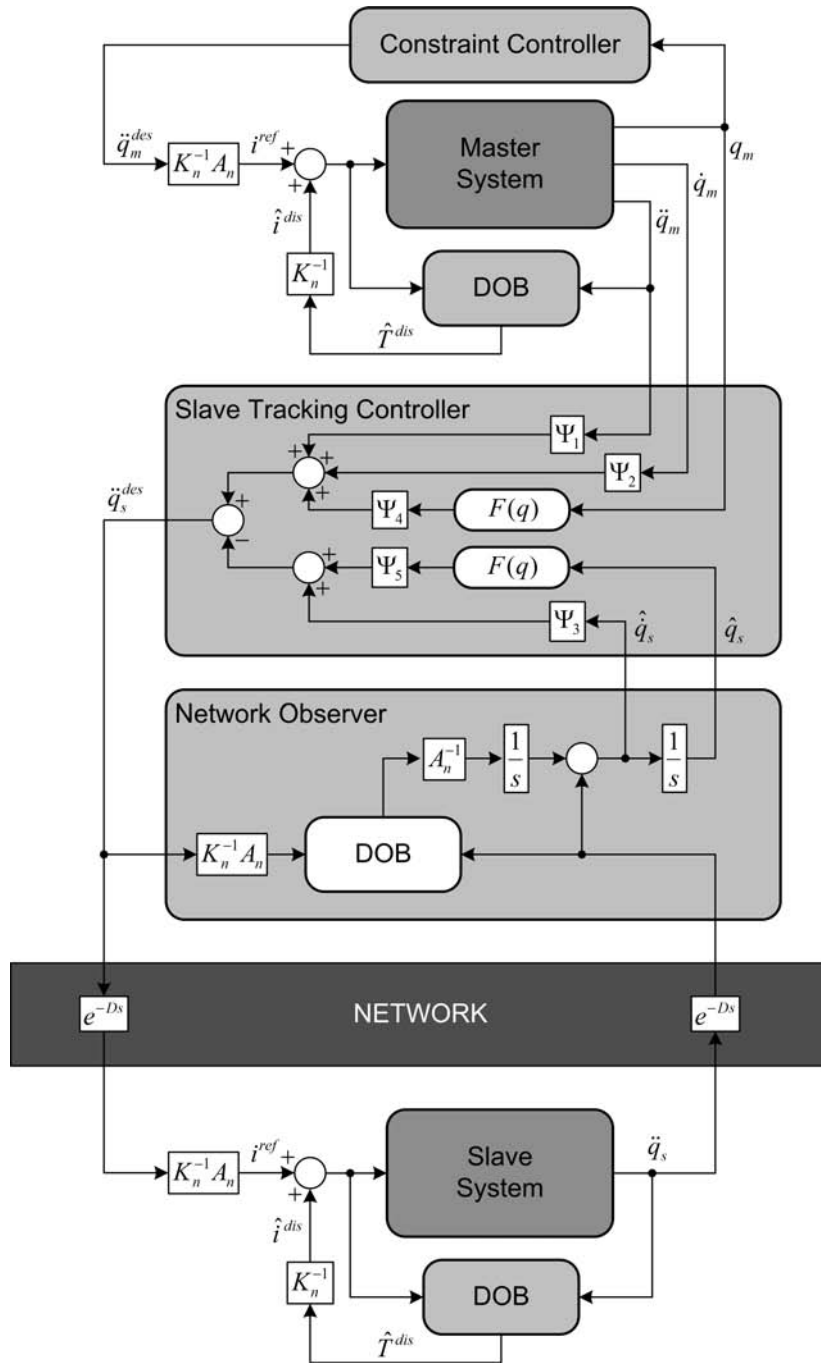


Figure 5.4: Structure of the Slave System Tracking Controller With Time Delay

Chapter VI

6 Realization

Realization of fast laser machining require utilization of fast actuators with as low inertia as possible. Besides the rapid operation capability, the end effector orientation of the slave actuator should necessarily preserve the perpendicularity with the laser beam. Moreover, requirement of free motion capability along constraining curve of master system enforces the possibility of back-drivability which prevents the use of gearboxes in the system. Considering these requirements, the most feasible solution to be used for both of master and slave systems comes out to be prismatically actuated direct drive parallel delta robots. Hence, in both systems linear delta structures with direct drive actuation is preferred and used.

Furthermore, in order to show the applicability of the methods proposed in the preceding chapters, mechanisms with different kinematic configurations is preferred for master and slave systems respectively. Being more specific, the master system contains a linear delta robot with horizontal alignment of actuators whereas in slave system a Keops structure (i.e. delta robot with actuators having certain non-zero attack angle) is preferred with smaller workspace than that of the master robot. Considering the differences of kinematic configurations between these two robots, in this chapter, a gener-

alized kinematic formulation is made for prismatically actuated delta robots. The derivation made below brings the advantage of applicability over many different linear delta mechanisms including the ones that does not preserve the symmetric ordering of motion axes.

6.1 Background

Direct kinematic formulation of parallel manipulators without requirement for iterative vector loop solution is still an open problem for many of the existing systems. Yet, there is not a well-defined methodological way of analysis to be applicable for the whole family of parallel robots. As a part of the study conducted in this thesis, a method for obtaining the direct forward and inverse kinematic solution of linear delta robot is developed. The method shows similarities to the derivations presented in [116] while preserving the flexibility to be applied for linear delta robots with asymmetric workspace. The derivation will refer to the geometry and variables (i.e. link lengths, points and angles) shown in Figure 6.1.

The derivation given below presents a novel approach for obtaining the position and motion level kinematics of prismatically actuated parallel delta type robots using a parametric geometric analysis. The contribution brought with the presented derivation enables easy kinematic calculation of linear delta robots that have different and potentially uneven configurations as shown in Figure 6.2 & Figure 6.3. In the analysis, the climbing angle between the base and the linear actuators and relative angles between each independent actuator plane are kept parametric. Hence, the derived model can be utilized for Horizontal [117], Vertical [118] and Keops [119] type of delta robots both for their even (i.e. actuator axes aligned with 120° angle differences) and

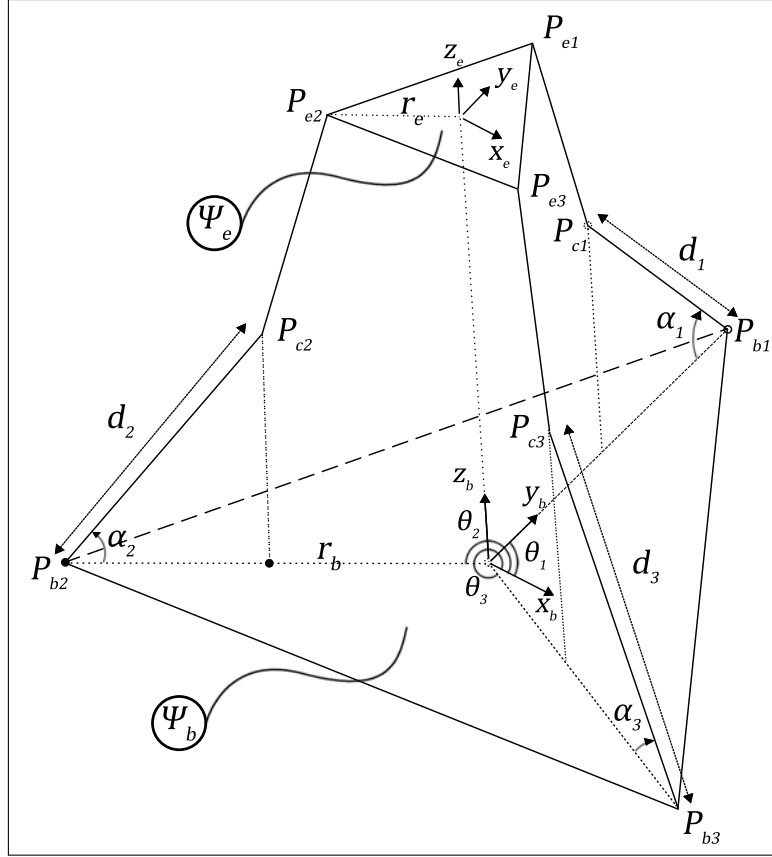


Figure 6.1: Schematic Representation of Delta Robot Geometry

uneven (i.e. actuator axes aligned with random angles) configurations.

6.2 Geometric Relationships

Let us assume that two coordinate frames, $\Psi_B = [x_b, y_b, z_b]^T$ and $\Psi_E = [x_e, y_e, z_e]^T$ are placed at the circumcenter of base and end effector triangles respectively. Here, it is implicitly assumed that the angles between any two links of the manipulator measured on the plane containing points P_{b1} , P_{b2} and P_{b3} is always below π radians. Hence, the circumcenter of the triangles always sit inside the area covered by the triangle plane. Using this representation,

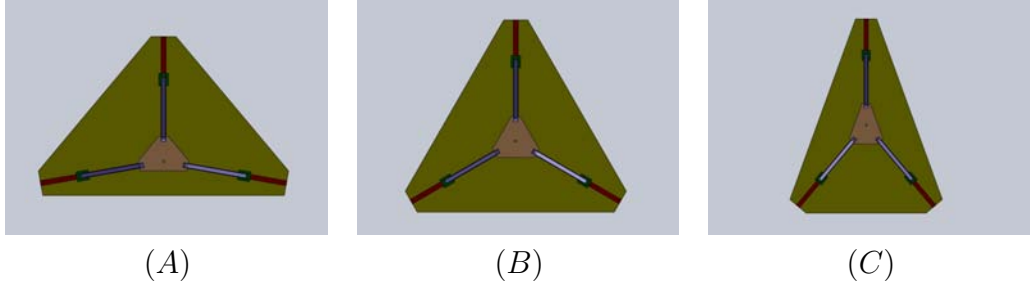


Figure 6.2: Variation of angle θ_i between base and axes of prismatic actuator projections: $\theta_1 = 90^\circ, \theta_2 = 190^\circ, \theta_3 = 350^\circ$ (A), Configuration with angles: $\theta_1 = 90^\circ, \theta_2 = 210^\circ, \theta_3 = 330^\circ$ (B), Configuration with angles: $\theta_1 = 90^\circ, \theta_2 = 230^\circ, \theta_3 = 310^\circ$ (C)

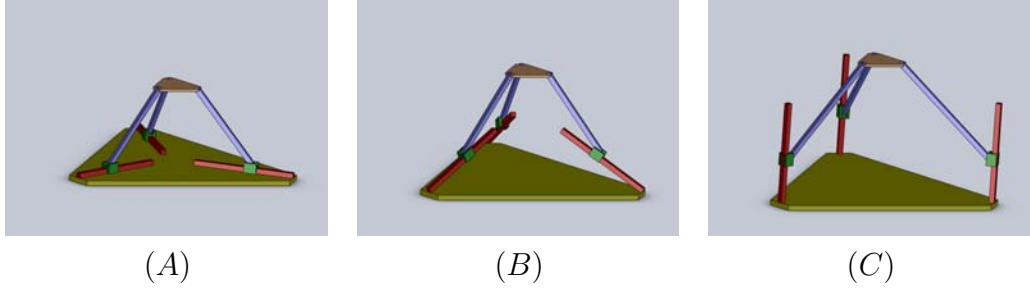


Figure 6.3: Variation of angle α_i between base and prismatic actuators. Horizontal Configuration ($\alpha_i = 0^\circ$) (A), Keeps Configuration ($\alpha_i = 30^\circ$) (B), Vertical Configuration ($\alpha_i = 90^\circ$) (C)

the position mapping between the two frames can be given as follows;

$$\Psi_e = \Psi_b + \Xi \quad (97)$$

where, $\Xi = [x, y, z]^T$ is the variable of interest in forward kinematics problem (i.e. the end effector coordinates with respect to origin set in the manipulator base). Following this definition, the coordinates of end effector triangle corners P_{ei} expressed with respect to base frame can be given by;

$$P_{ei} = \Psi_e + R_{z,\theta_i} T_e \quad (98)$$

where, $R_{z,\theta_i} \in SO(3)$ and $T_e = [r_e, 0, 0]^T$ stand for the rotation matrix around z-axis by an angle of θ_i and translation vector along x-axis by a magnitude of r_e . In the most generalized case, variation of angle θ_i would enable generation of asymmetric geometries for the manipulator, which will bring designing flexibility based on the desired precision requirements for certain axes of the end effector. Some examples of such uneven configurations are shown in Figure 6.2. Substitution of equation (97) into equation (98) yields the following identity;

$$P_{ei} = \Psi_b + \Xi + R_{z,\theta_i}T_e \quad (99)$$

Similarly, the coordinates of the base triangle corners P_{bi} can be given as;

$$P_{bi} = \Psi_b + R_{z,\theta_i}T_b \quad (100)$$

where, $T_b = [r_b, 0, 0]^T$ represent the translation vector along x-axis by a magnitude of r_b . Assuming that each motor i realizes a translation of magnitude d_i along its axis of motion, one can represent the coordinates of the tip of motors P_{ci} expressed in the base frame as follows;

$$P_{ci} = \Psi_b + R_{z,\theta_i}T_b + R_{z,\theta_i}R_{y,\phi_i}T_{di} \quad (101)$$

where, $T_{di} = [0, 0, d_i]^T$ and $\phi_i = \alpha_i - \pi/2$ stand for the translation vector along z-axis by a magnitude of d_i and for the counter-clockwise rotation angle of i^{th} actuator around its own y-axis respectively. Keeping a parametric value for the angle α_i (usually referred as the attack angle of i^{th} joint) brings further designing flexibility based on the requirements desired from the manipulator. Some example configurations are illustrated in Figure 6.3.

Since the definitions of the points representing the system's geometry are all made with respect to the base axis, one can now insert the final constraint and formulate the fundamental equations for forward and inverse kinematics of the system. Assuming that the links between points P_{ci} and P_{ei} have the lengths l_i , the constraining equation of the system can be given by;

$$\|P_{ei} - P_{ci}\| = l_i \quad (102)$$

where, $\|\cdot\|$ represent the L_2 norm of its argument. Substituting equations (98) and (101) to equation (102), one can obtain the following identity;

$$\|\Xi + R_{z,\theta_i}T_e - R_{z,\theta_i}T_b - R_{z,\theta_i}R_{y,\phi_i}T_{di}\| = l_i \quad (103)$$

Equation (103) can be recast by substituting the contents of vectors Ξ , T_e , T_b , and the matrices R_{z,θ_i} , R_{y,ϕ_i} as follows;

$$\left\| \begin{bmatrix} x + (r_e - r_b + d_i c\alpha_i) c\theta_i \\ y + (r_e - r_b + d_i c\alpha_i) s\theta_i \\ z - d_i s\alpha_i \end{bmatrix} \right\| = l_i \quad (104)$$

where, c^* and s^* stand for the abbreviations of the cosine and sine of the angle $*$. Expanding the norm in (104), one can finally obtain the following quadratic equation;

$$\begin{aligned} (x + c\theta_i (r_e - r_b + d_i c\alpha_i))^2 + (y + s\theta_i (r_e - r_b + d_i c\alpha_i))^2 + \\ (z - d_i s\alpha_i)^2 = l_i^2 \end{aligned} \quad (105)$$

This equation sets the baseline for derivation of both position and velocity

level kinematics.

6.3 Position Level Kinematics

Equation (105) for $i = 1, 2, 3$ contain three equations in three unknowns for each case of forward and inverse kinematics. In forward kinematics, lengths d_i are known and the unknowns of the equations are the end effector positions $[x, y, z]^T$ while in inverse kinematics the lengths d_i are unknowns to be calculated from the given end effector positions $[x, y, z]^T$.

6.3.1 Forward Kinematics

The solution to the forward kinematics problem geometrically give the intersection points of three spheres with the corners of a triangle that is parallel to the base plane. Recalling from the structure given in Figure 6.1, this intersection can as well be at a point which corresponds to the lower arm configuration of the delta robot. Hence, there exists two solutions for each of x , y and z . Having known that, we can now continue with the solution of the forward kinematics problem.

Referring back to equation (105), one can rearrange the terms and come up with the following expression;

$$x^2 + y^2 + z^2 + \sigma_{1i}x + \sigma_{2i}y + \sigma_{3i}z - \lambda_i = 0 \quad (106)$$

where, the terms σ_{1i} , σ_{2i} , σ_{3i} and λ_i has the following content;

$$\sigma_{1i} = 2c\theta_i (d_i c\alpha_i + r_e - r_b) \quad (107)$$

$$\sigma_{2i} = 2s\theta_i (d_i c\alpha_i + r_e - r_b) \quad (108)$$

$$\sigma_{3i} = -2d_i s\alpha_i \quad (109)$$

$$\lambda_i = l_i^2 - d_i^2 - (r_e - r_b)^2 - 2d_i r_e c\alpha_i + 2d_i N c\alpha_i \quad (110)$$

The expression given in (106) contain three linearly independent equations. All of these equations include both first and second order terms which makes the set difficult to solve. On the other hand, all of the quadratic terms in these equations have the same coefficient and can be eliminated by subtraction. Making use of (106) one can subtract the hidden equations for $i = 1, 2, 3$ from one another and come up with the following equation system:

$$(\sigma_{11} - \sigma_{13})x + (\sigma_{21} - \sigma_{23})y + (\sigma_{31} - \sigma_{33})z = (\lambda_1 - \lambda_3) \quad (111)$$

$$(\sigma_{12} - \sigma_{13})x + (\sigma_{22} - \sigma_{23})y + (\sigma_{32} - \sigma_{33})z = (\lambda_2 - \lambda_3) \quad (112)$$

Equations (111) and (112) can be recast as a linear matrix equality as follows;

$$A\Upsilon = \Gamma z + \Lambda \quad (113)$$

where we have;

$$A = \begin{bmatrix} \sigma_{11} - \sigma_{13} & \sigma_{21} - \sigma_{23} \\ \sigma_{12} - \sigma_{13} & \sigma_{22} - \sigma_{23} \end{bmatrix} \quad (114)$$

$$\Upsilon = \begin{bmatrix} x \\ y \end{bmatrix} \quad (115)$$

$$\Gamma = \begin{bmatrix} \sigma_{33} - \sigma_{31} \\ \sigma_{33} - \sigma_{32} \end{bmatrix} \quad (116)$$

$$\Lambda = \begin{bmatrix} \lambda_1 - \lambda_3 \\ \lambda_2 - \lambda_3 \end{bmatrix} \quad (117)$$

The solution for equation (113) can be obtained by a matrix inversion and can be given as;

$$\Upsilon = A^{-1}\Gamma z + A^{-1}\Lambda \quad (118)$$

Equation (118) can give the expression for Υ if and only if the matrix A is invertible. The invertibility of matrix A is analyzed and proven in Appendix B at the end of the manuscript. Having proven the invertibility of matrix A , we can now further proceed with the solution of equation (106). Defining the vectors;

$$u = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (119)$$

$$v = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (120)$$

and making use of equation (118) one can obtain the expression for x and y in terms of z as follows;

$$x = uA^{-1}\Gamma z + uA^{-1}\Lambda \quad (121)$$

$$y = vA^{-1}\Gamma z + vA^{-1}\Lambda \quad (122)$$

Without loss of generality, one can rewrite equations (121) and (122) as

follows;

$$x = g_1 z + e_1 \quad (123)$$

$$y = g_2 z + e_2 \quad (124)$$

where, $\{g_1, g_2, e_1, e_2\} \in \mathbb{R}^{1 \times 1}$. Having the expression of x and y , one can now make the substitution back to any one of the equation (106). Picking the first one, we can recast the following quadratic equation;

$$(g_1 z + e_1)^2 + (g_2 z + e_2)^2 + z^2 + \sigma_{11}(g_1 z + e_1) + \sigma_{21}(g_2 z + e_2) + \sigma_{31}z - \lambda_1 = 0 \quad (125)$$

The terms in equation (125), can be grouped and the following simplified expression for z can be obtained;

$$\eta z^2 + v z + \delta = 0 \quad (126)$$

where, we have;

$$\eta = g_1^2 + g_2^2 + 1 \quad (127)$$

$$v = 2e_1 g_1 + 2e_2 g_2 + \sigma_{11} g_1 + \sigma_{21} g_2 + \sigma_{31} \quad (128)$$

$$\delta = e_1^2 + e_2^2 + \sigma_{11} e_1 + \sigma_{21} e_2 - \lambda_1 \quad (129)$$

The solution to the system given in (126) can be obtained from the discriminant analysis. The two roots can directly be written as;

$$z_{1,2} = \frac{-v \pm \sqrt{v^2 - 4\eta\delta}}{2\eta} \quad (130)$$

Here, the two roots represent the upper and lower configuration of the delta robot discussed before. In order to have the correct value for z , we have to take the root;

$$z = \frac{-v + \sqrt{v^2 - 4\eta\delta}}{2\eta} \quad (131)$$

Once z is obtained, we can further proceed to calculate x and y from equations (123) and (124) respectively, which concludes the forward kinematic calculations.

In order to summarize the procedure for the forward kinematics calculation, below an algorithmic explanation is given which provides a recipe for practical realization.

Algorithm 2 Summary of Forward Kinematics

Given the system configuration parameters α_i and θ_i and actuator displacements d_i for $i = \{1, 2, 3\}$;

- Calculate $\sigma_{1i}, \sigma_{2i}, \sigma_{3i}, \lambda_i$
 - Form matrices A, Γ, Λ
 - Calculate g_1, g_2, e_1, e_2
 - Calculate η, v, δ
 - $z = \frac{-v + \sqrt{v^2 - 4\eta\delta}}{2\eta}$
 - $y = g_2 z + e_2$
 - $x = g_1 z + e_1$
-

6.3.2 Inverse Kinematics

Like forward kinematics, equation (105) also constitute the basis for the derivation of inverse kinematics. However, now the positions x , y and z are known while the displacements of linear actuators d_i are unknown. Rear-

ranging this equation for the unknown terms, one can obtain the following expression;

$$d_i^2 + \mu_{1i}d_i + \mu_{2i} = 0 \quad (132)$$

where, the terms μ_{1i} and μ_{2i} have the following content;

$$\mu_{1i} = 2c\alpha_i (r_e - r_b + c\theta_i x + s\theta_i y) - 2s\alpha_i z \quad (133)$$

$$\mu_{2i} = -L^2 + x^2 + y^2 + z^2 + (r_e - r_b) (r_e - r_b + 2c\theta_i x + 2s\theta_i y) \quad (134)$$

Expression given in (132) contain three quadratic equations each of which contain single unknown d_i for $i = \{1, 2, 3\}$ respectively. Further solution of these quadratic equations can be obtained using discriminant analysis as follows;

$$\Delta_i = \mu_{1i}^2 - 4\mu_{2i} \quad (135)$$

Recalling the solution of quadratic systems, we will have two roots using the Δ_i s above and they are given as;

$$d_{i1,2} = \frac{-\mu_{1i} \pm \sqrt{\Delta_i}}{2} \quad (136)$$

It is important to take the correct root from the system. As stated earlier, under the condition that $\|\cdot\|$ represents L_2 norm, equation (102) geometrically reflects a sphere equation. Unlike forward kinematics case, however, in inverse kinematics the center of the sphere is assumed to be located at point P_{ei} . Originating from that, the feasible solution among the two solutions of equation (132) will be the one that has lower magnitude. Hence, the solution

of inverse kinematics problem can be given as;

$$d_i = \frac{-\mu_{1i} - \sqrt{\Delta_i}}{2} \quad (137)$$

Like many other parallel manipulators, the inverse kinematics calculations are simpler than the forward kinematics. For the sake of completeness, below the algorithmic procedure of inverse kinematics calculations is given.

Algorithm 3 Summary of Inverse Kinematics

Given the system configuration parameters α_i and θ_i for $i = \{1, 2, 3\}$ and end effector coordinates $\{x, y, z\}$;

- Calculate μ_{1i}, μ_{2i}
 - Calculate Δ_i
 - $d_i = \frac{-\mu_{1i} - \sqrt{\Delta_i}}{2}$
-

6.4 Velocity Level Kinematics

In order to obtain the velocity level kinematics of the system being analyzed, one can again refer to the constraining equation given in (105). Differentiating that equation, one can obtain

$$f_{di}(t)\dot{d}_i(t) = f_{xi}(t)\dot{x}(t) + f_{yi}(t)\dot{y}(t) + f_{zi}(t)\dot{z}(t) \quad (138)$$

where, the functions $f_{xi}(t)$, $f_{yi}(t)$, $f_{zi}(t)$ and $f_{di}(t)$ for $i = \{1, 2, 3\}$ have the following content;

$$f_{xi}(t) = c\theta_i(r_b - r_e - c\alpha_i d_i(t)) - x(t) \quad (139)$$

$$f_{yi}(t) = s\theta_i(r_b - r_e - c\alpha_i d_i(t)) - y(t) \quad (140)$$

$$f_{zi}(t) = s\alpha_i d_i(t) - z(t) \quad (141)$$

$$f_{di}(t) = c\alpha_i(r_e - r_b + c\theta_i x(t) + s\theta_i y(t)) - s\alpha_i z(t) + d_i(t) \quad (142)$$

Equation (138) represent three embedded equations for $i = \{1, 2, 3\}$. The objective of motion level kinematics is to come up with the mapping between the joint (i.e. configuration) space velocities $\dot{\mathbf{q}}(t) = [\dot{d}_1(t), \dot{d}_2(t), \dot{d}_3(t)]^T$ and the task (i.e. operational) space velocities $\dot{\mathbf{x}}(t) = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]^T$. To formulate that mapping, one can diagonalize the terms $f_{di}(t)$ of equation (142) and come up with the following identity;

$$J_1 \dot{\mathbf{q}}(t) = J_2 \dot{\mathbf{x}}(t) \quad (143)$$

where, the contents of matrices J_1 and J_2 are given as;

$$J_1 = \begin{bmatrix} f_{d1}(t) & 0 & 0 \\ 0 & f_{d2}(t) & 0 \\ 0 & 0 & f_{d3}(t) \end{bmatrix} \quad (144)$$

$$J_2 = \begin{bmatrix} f_{x1}(t) & f_{y1}(t) & f_{z1}(t) \\ f_{x2}(t) & f_{y2}(t) & f_{z2}(t) \\ f_{x3}(t) & f_{y3}(t) & f_{z3}(t) \end{bmatrix} \quad (145)$$

Making use of equation (143), one can write down the kinematic jacobian J_x

of the system as;

$$J_{\mathbf{x}} = J_2^{-1} J_1 \quad (146)$$

The Jacobian Matrix derived in equation (146) is very important for the practical applications over the manipulator under consideration, since both of the force/torque and velocity mappings between the configuration and operational spaces are carried out using it. The existence of Jacobian Matrix is dependent on the non-singular structure of matrix $J_{\mathbf{x}}$. This matrix will be singular in case that any of the matrices J_2 or J_1 is rank deficient. The rank analysis of these matrices require the particular structure of the delta robot under consideration be known (i.e. exact information about angles α_i , θ_i and lengths l_i) and is beyond the context of this study.

Chapter VII

7 Experimental Platform & Experiment Specific Algorithms

In order to present a full picture of the methodology discussed in the preceding chapters, the derived algorithms and controllers are tested on a multi-DOF experimental platform. This platform consists of two 3-DOF prismatically actuated delta robots serving as the master and slave systems respectively, one laser unit mounted on top of the slave system for machining purposes, one visual inspection system used for the acquisition of images, one computer and a digital signal processing unit. In the discussion given below, first, details related to the components of the experimental platform are presented. Once a general picture of the overall system is presented, the specific algorithms (either proposed or imported from existing literature) used for the realization of the experiments are discussed.

7.1 Components of Experimental Platform

In this section, the components that constitute the overall experimental platform is discussed in terms of their operational specifications. In that sense, mechanical structure and properties of manipulators and visual inspection

unit, machining specifications of laser cutting unit and specifications of experiment computer and real time processing unit are explained in detail.

7.1.1 Master System

For the master robot platform, a prismatically actuated "horizontal" type delta robot is designed and produced. Due to the requirement that master robot is supposed to enforce constraints to the operator, the force scale of the actuators and correspondingly the dimensions of the robot is selected to be relatively larger. Moreover, due to larger force generation ability of the setup, in the master robot, precise roller bearings are used as the joint connection units.

The robot contains three Faulhaber[©] linear brushless DC motors from LM2070 series with integrated hall effect sensor readouts enabled for use in their corresponding drivers. The motors are equipped with incremental encoders from Renishaw[©] that have resolution of $0.5\mu m$. The attack angle is adjusted to have a value of 0° due to the horizontal structure. Each motor has a maximum continuous force rating of $10N$. Considering the equilateral triangular structure of the setup, a maximum of $20N$ force can be implied on the end effector. Furthermore, each motor has an averaged inertia of $335g$ and the end effector has a total mass of $164g$. A picture of the produced setup is given in Figure 7.1.

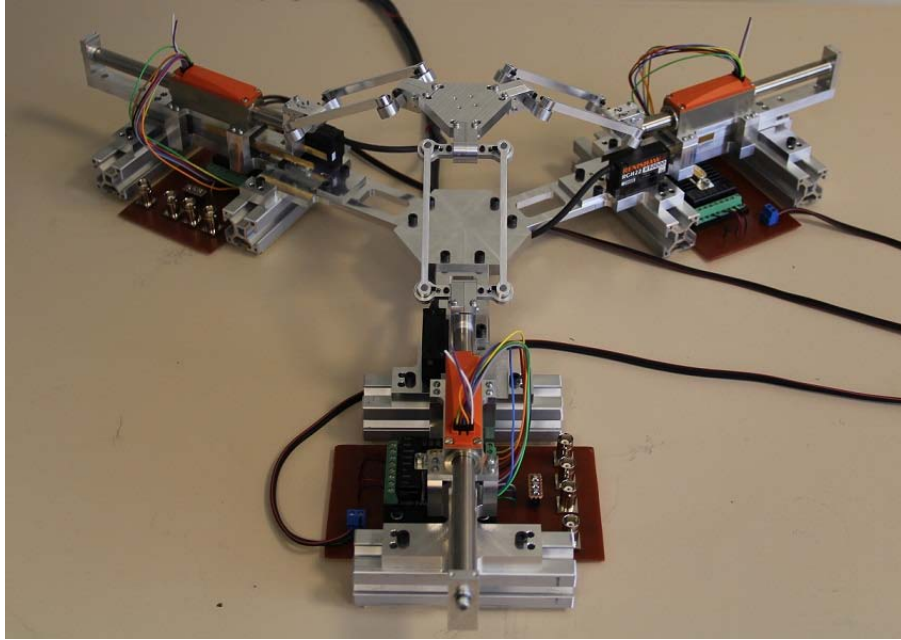
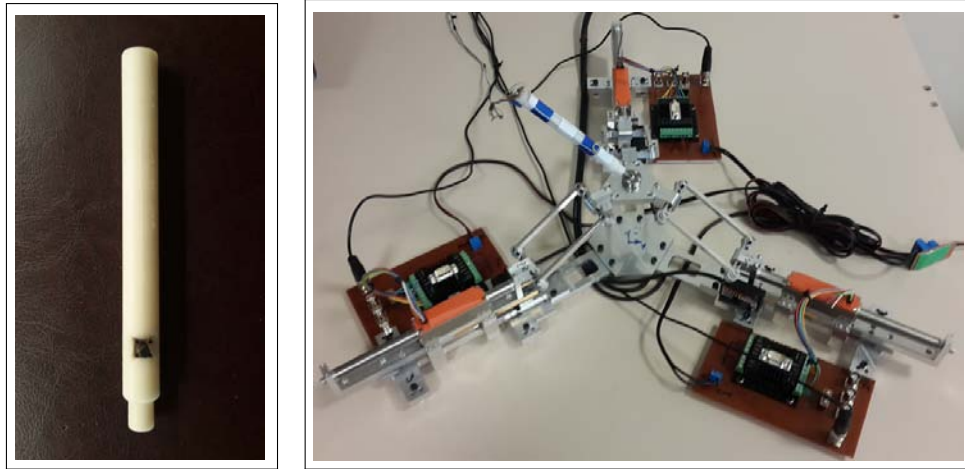


Figure 7.1: The Horizontal Linear Delta Setup (Master Robot)

In order to provide the human operator with ability of remote laser marking on the master system, a laser marking pen is designed and attached to the tip of the master system end effector via a high precision universal joint. The laser marking pen is first designed in a CAD program and then produced in a 3D printer. In order to enable laser shooting, a button is attached over the produced pen. Depiction of the produced laser marking pen and the master system with the attached pen are shown in Figure 7.2-(A) and Figure 7.2-(B) respectively.



(A)

(B)

Figure 7.2: Master System; Laser Marking Pen (A), Master Robot with Attached Marking Pen (B)

7.1.2 Slave System

As the slave robot platform, a miniature "Keops" type prismatically actuated delta robot is designed and produced. Since the slave system has only the assignment of position tracking without interaction force from environment, low force actuators are preferred in the design. Due to low weight requirements, a new joint design replacing the conventional joint structures such as bearings was made for this robot. This new joint structure is based on the use of very small, hard, ruby ball-probe pairs produced by Renishaw[©] company. The balls are placed between specially produced aluminum bars with hemispherical cross-section. The design of the aluminum bars are made in such a way that the ruby piece can move easily and does not leave gap at the contact points. Due to very low friction between aluminum and ruby, the sandwiched balls can take place of universal joints and be attached to

the assembly by the threaded ends of the probes. This way, a total of 24 steel bearings are replaced with 12 probes which considerably decreased the moving system weight. A depiction of the joint structure is given below in Figure 7.3. In the final design, the end effector of the platform has a total mass of $68g$.

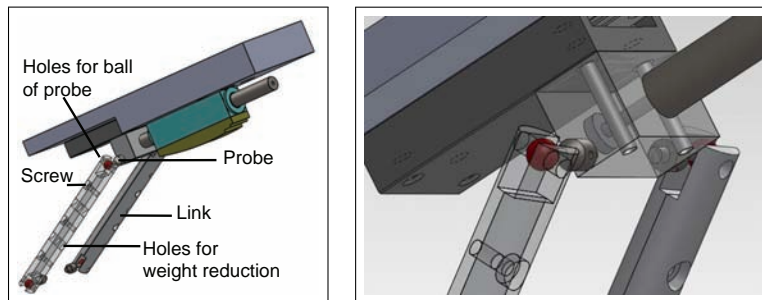


Figure 7.3: Joint Design Used in the Slave System

Besides the joints, the robot contains three Faulhaber[©] linear brushless DC motors from LM1247 series with integrated hall effect sensor readouts enabled for use in their corresponding drivers. These hall effect readouts are also received by the computer through ADC connectors for calculation of motor positions. The hall effect position sensing algorithm is described in the following sections. The maximum continuous force rating of each motor is $3.6N$ with an inertia of $71g$. The attack angle (i.e. the climbing angle of Keops) is adjusted to have a value of 30° . A picture of the produced setup is given in Figure 7.4

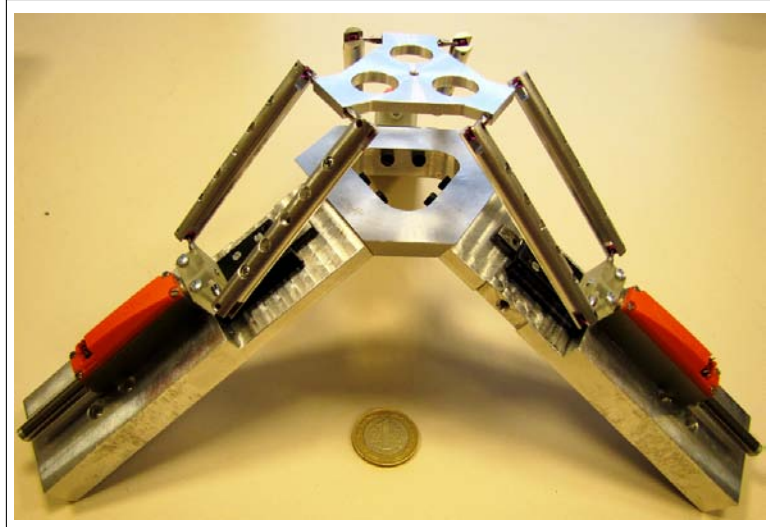


Figure 7.4: The Keops Linear Delta Setup (Slave Robot)

7.1.3 Laser Machining Unit

Laser machining unit consists of a workstation that contains a laser head and the corresponding mechanical linkage to hold this laser head and move it in vertical axis for the coarse adjustment of focus point. The laser system used in the workstation is SPI red ENERGY G3 HS series pulsed fiber laser capable of delivering nanosecond pulses with the repetition rates up to 500 KHz. The laser beam is created by a laser generation system mounted at a distant location than the workstation and transferred to the machining unit via a fiber optic connection. The laser generation system consists of several modules that include beam generation unit, laser controller and isolator, heat sink, internal red alignment laser (RAL), power supply, breakout board and corresponding electronics. The generated laser beam has 1065nm wavelength and a maximum operating power of 20 watts. The base of the laser machining unit is aligned perpendicular to the laser shooting direction for

the attachment of slave system used in the machining operation. For better understanding of the overall workstation, a depiction of the CAD drawing of laser machining unit is given below in Figure 7.5 while the actual system can be seen in Figure 7.9.

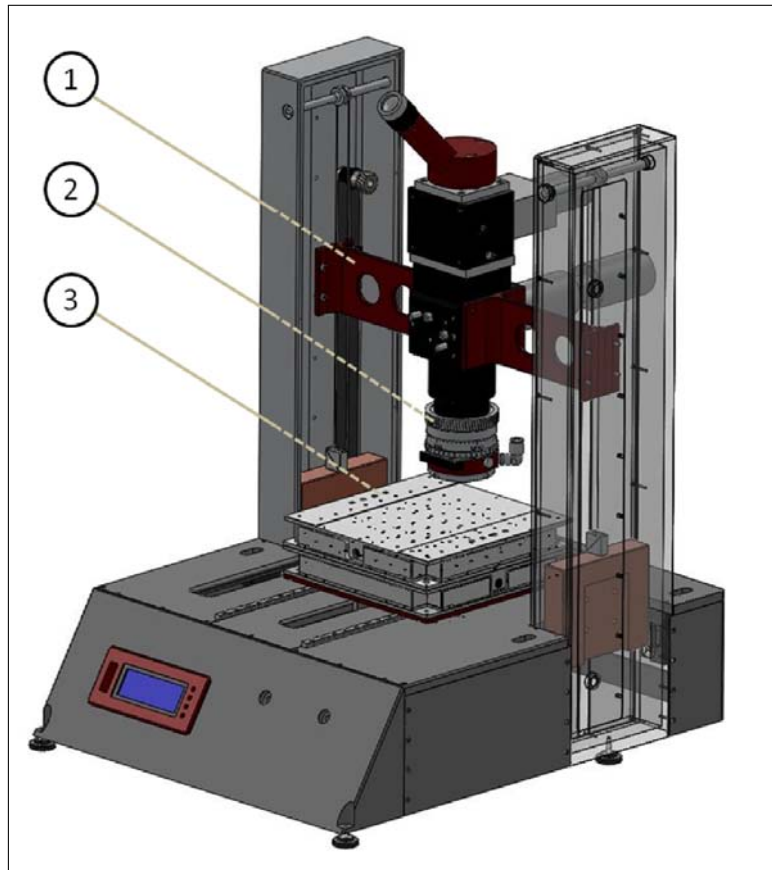


Figure 7.5: Illustration of Laser Machining Unit: Vertical Adjustment Sled ①, Laser Head ②, Robotic Manipulator ③

7.1.4 Image Acquisition Unit

In order to acquire images of the contour to be machined, a visual inspection unit is designed and produced. The presented system has two degrees

of freedom to change magnification and focus of a camera integrated microscope system respectively. Following the design and production of this image acquisition unit, the relationship between the focus level and the magnification (the effect of changing magnification on focus and changing focus on magnification) is analyzed and a self optimizing controller is proposed based on acceleration control framework.

In order to have a miniature structure that can easily be embedded into the experimental setup, the size of the selected microscope should also be small. In that sense, among the existing suppliers, TIMM 400 Digital camera integrated miniature microscope is selected as the core of the image acquisition system. The selection is made based on the capabilities provided by the mentioned system and the desired specifications for potential applications. The selected microscope & camera system has an integrated USB plug through which the user can gather the image data. A picture of the selected microscope is given in Figure 7.6 below.

The working structure of image acquisition system provides two degrees of freedom (DOF) which have to be adjusted simultaneously to have the desired images. Magnification level of the system is primarily related to the distance between the microscope and the object of interest to be viewed. On the other hand, the focusing screw as shown in Figure 7.6 is responsible for the adjustment of focus point.

In order to have the capability to adjust both magnification and focus, a setup is designed that is supposed to hold and move the microscope body and focusing unit by two DC motors respectively. For the up down movement of system, a ball screw driven by a relatively more powerful motor is preferred since the ball screw structure brings the advantage of self locking in exchange

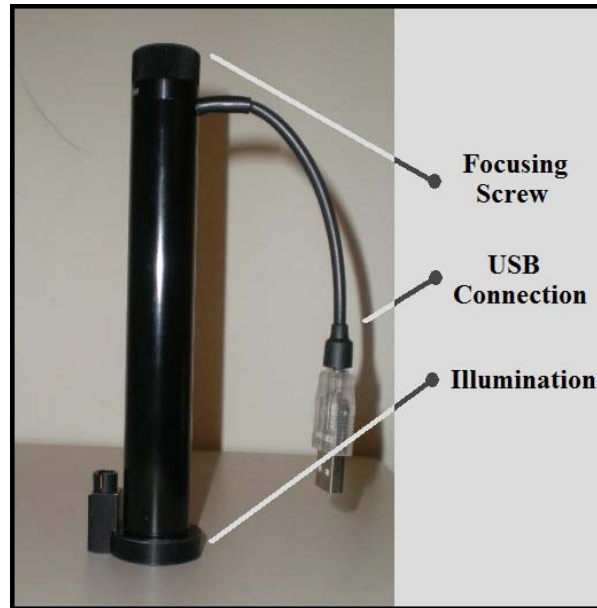


Figure 7.6: TIMM 400 Digital Series Microscope with Integrated Camera

of bigger friction force during motion. On the other hand, the focusing screw of the image acquisition unit is driven by a relatively smaller force motor since there is almost negligible force over the rotary motion of that screw. The CAD drawing and the produced setup is shown in Figure 7.7. During the design process of the setup, several important aspects related to the precision of the system in real time experiments are considered.

As the actuation system, two Faulhaber DC motors are used from series 22/5 and 15/8 for the magnification and focusing motions respectively. Both of these motors are equipped with zero backlash gearheads and high precision optical encoders. In order to provide a smooth vertical motion, the microscope is held at the back with a linear bearing from Schneeberger NDN2-80.70 series. Finally, the motors are attached to the corresponding screws (i.e. ball screw for vertical motion and focusing screw of camera system for focusing

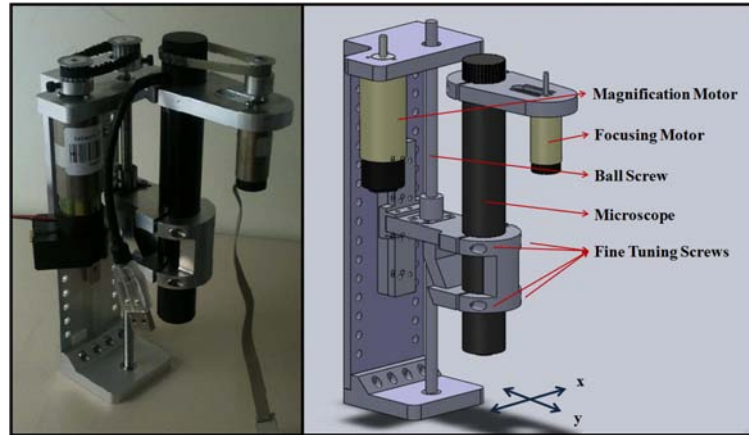


Figure 7.7: Produced Image Acquisition Setup and its CAD Drawing

motion) with tightened pulley and belt systems.

7.1.5 Intermediate Components

In order to realize the derived algorithms, several intermediate components are also integrated to the experimental platform. Among those components; a real-time processing mother board, PCB boards for drive electronics, a supervisory computer and the communication interface between master and slave system exist. Below, the details related to these intermediate components are given.

The real-time processing is enabled via a DSpace DS1103 mother board. The processor board is equipped with one 50 bit-I/O channel, 36 A/D channels, 8 D/A channels and 7 channels for incremental encoder readouts. The algorithms are coded in C environment and compiled over the platform. The coded algorithms are optimized to a certain extent using a software framework. This way, in full functional configuration of the setup, the maximum achievable loop frequency could be boosted up to 2.4 KHz.

For the purpose of having neat response and clean measurement, PCB boards are designed and produced for the drive electronics of experimental platform. Particularly, the analog hall effect sensor measurement noise is reduced via the utilization of these PCB boards.

In order to monitor and interrupt the real-time process being carried out in the platform, a supervisory computer is used. The supervisory computer consists a single core Pentium-4 series processor with 2.2 GHz clock frequency and a 2GB of RAM. The graphical user interface used for monitoring is the one provided with the software package of DSpace mother board.

The last intermediate component used in the platform is the communication medium between master and slave systems. Network environment is used for communication between the systems. For the validation of remote machining experiments, a secondary computer is used with the slave system and the Transmission Control Protocol (TCP) is set between the two computers. Further, the behavior of TCP communication over network (i.e. delay with constant and variable components and no data loss) is coded in the system and the secondary computer is removed. The results for remote non-contact machining experiments are obtained using this coded network structure (i.e. having artificially generated delays between master and slave systems).

Depictions of these intermediate components are given below in Figure 7.8 below.

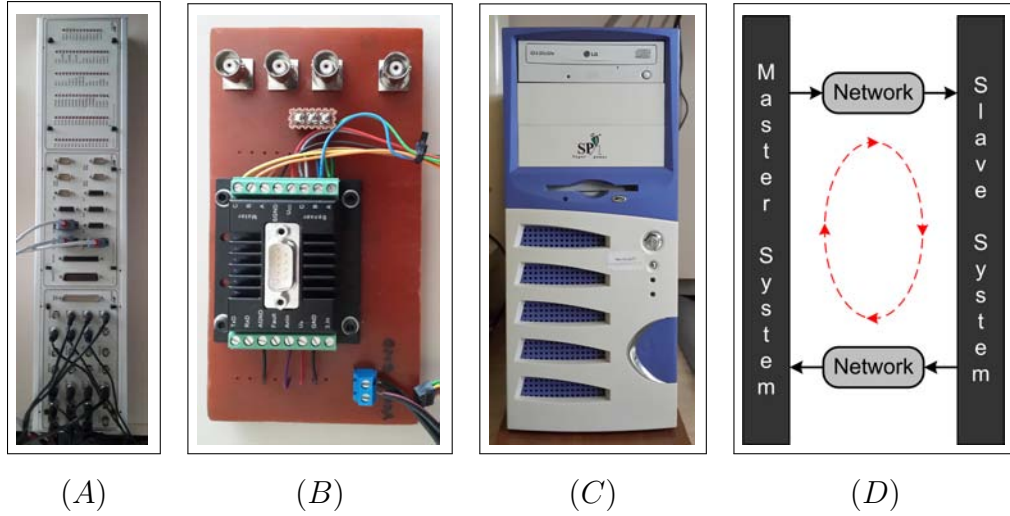


Figure 7.8: Intermediate Components Used for Experimental Verification; Real-Time Processing Unit (A), Driver Electronic Boards (B), Supervisory Computer (C), Communication Medium (D)

7.1.6 Overall System

The components mentioned above are all produced and combined in a functional way to give the overall experimental platform. Below, a depiction of the overall experimental platform is given in Figure 7.9. In that figure, the components include; laser machining unit ①, slave robot ②, real-time processing unit ③, process monitoring unit ④, image acquisition unit ⑤, master robot ⑥ and the power source ⑦.

7.2 Algorithms & Methods Utilized for Experimental Realization

Besides the components of experimental platform, in this section, experiment specific algorithms developed for the realization purposes are also explained.

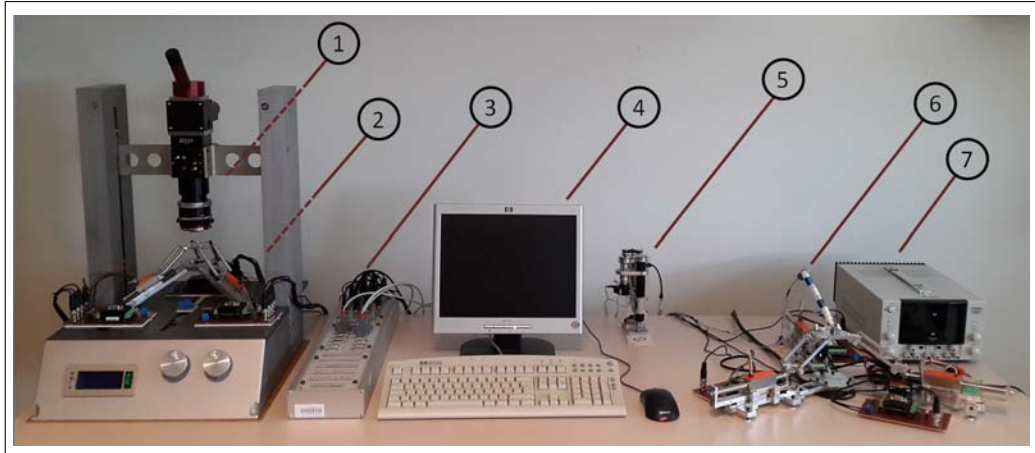


Figure 7.9: Overall Experimental System

Among these algorithms; a scheme for analog position measurement of linear motors, a new autofocusing algorithm for precise acquisition of image references and a novel compression method related to data transmission over network are presented.

7.2.1 Analog Position Measurement from Hall Effect Sensors

The direct drive linear actuators used in the system also provide an advantage in terms of position measurement. The principle behind linear motors rely on the three phase electromagnetic actuation technology. Hence, the direct drive prismatic actuators by default contain hall effect sensors in the motor housing to measure the location of the shaft. During the design of manipulator, this property of the brushless linear direct current (BLDC) motors are taken into consideration for position measurement of the shaft without the requirement of installing an external incremental encoder.

Estimation of position from analog hall effect sensor readouts is recently popularized in practical applications by some researchers [120]. The methodol-

ogy relies on the fact that the shaft of the motor contains aligned permanent magnets with magnetic pitch length τ_m and the motor housing contains 3 magnetic sensors separated from each other by a distance of $\tau_m/3$ radians. Hence, the voltage outputs from the sensors (hereby referred as u_1 , u_2 and u_3) are $2\pi/3$ radians phase shifted with respect to the preceding one. Mathematically, the content of the signals can be represented as;

$$u_1 = \sin(\phi) \quad (147)$$

$$u_2 = \sin(\phi + 2\pi/3) \quad (148)$$

$$u_3 = \sin(\phi - 2\pi/3) \quad (149)$$

Since the voltages vary by an angle of 2π radians over a distance equal to τ_m , the angle ϕ can be given as;

$$\phi = \frac{2\pi d}{\tau_m} \quad (150)$$

where, $d \leq \tau_m$ is the distance traveled by the corresponding motor shaft. In order to determine d , the signals u_1 , u_2 and u_3 are first transformed to orthogonal coordinate system (u_a, u_b) using the Clarke transformation as follows;

$$u_\alpha = \frac{2}{3} \left(u_1 - \frac{1}{2}u_2 - \frac{1}{2}u_3 \right) \quad (151)$$

$$u_\beta = \frac{2}{3} \left(\frac{\sqrt{3}}{2}u_2 - \frac{\sqrt{3}}{2}u_3 \right) \quad (152)$$

Once the signals are mapped into orthogonal coordinates, the value of d can

be obtained by;

$$d = \frac{\tau_m}{2\pi} \text{atan2}(u_\alpha, u_\beta) \quad (153)$$

During the implementation, analog signals are imported from analog to digital converters of a DSP board that has 16 bits quantization. Having known that the pitch length τ_m of the motors used in the system is $18mm$, one can calculate the resolution in the position measurement as;

$$R_x = \frac{\tau_m}{2^{16}} \approx 0.275\mu m \quad (154)$$

which is well beyond the resolution of many commercially available encoder systems.

7.2.2 A Novel Autofocusing Method for Sharp Image Acquisition

Following the production of the image acquisition setup, several tests are made to see the performance of the whole system. During the tests, it is observed that the independent motions to change focus and magnification of the system also affect the magnification and focus respectively. In other words, playing with focus affects the magnification and playing with magnification affects the focus. For the purpose of better illustration of these coupled effects, the experiment results that include motion only in the magnification motor and motion only in the focusing motor is presented below in Figure 7.10 and Figure 7.11 respectively.

In order to understand the coupling between the two motions, the system is programmed to have a full scan of the entire range of motion with two motors. For this purpose, the full motion range of focusing and magnification motors are divided into incremental motion segments. At each segment of

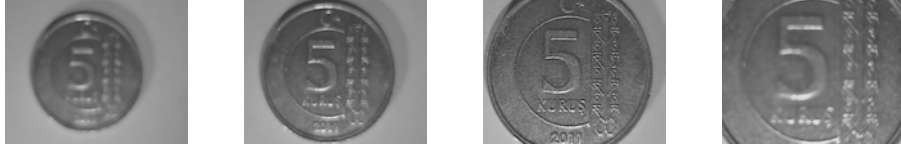


Figure 7.10: Magnification Sweep with the Image Acquisition System Over a 0.05 TRY coin

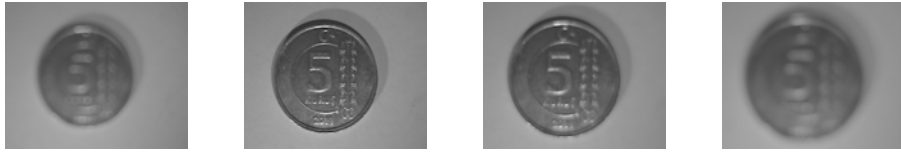


Figure 7.11: Focusing Sweep with the Image Acquisition System Over a 0.05 TRY coin

magnification motor, the focusing motor is programmed to sweep over all of its motion increments. Applying the same methodology for the entire range of motion segments of magnification motor, a 2D data set is acquired over which the information is given in terms of the sharpness. In order to have a better judgement of sharpness, Normalized Variance operator [121] is taken into consideration during the experiments. This operator has almost the same procedure as the Variance method [122]. But, rather than taking direct variance, this algorithm normalizes the sharpness values with the mean value of the intensity. With this procedure, the differences on consecutive images are diminished, which helps in getting a smoother behavior. The equation for Normalized Variance sharpness measure is given as:

$$F_{Norm. Var} = \frac{1}{H \cdot W \cdot \mu} \sum_{Height} \sum_{Width} [i(x, y) - \mu]^2 \quad (155)$$

where, H , W , $i(x, y)$ and μ are the number of pixels through height, number of pixels through width, the gray level intensity of the single pixel located at

(x, y) coordinates and the mean intensity of the image respectively.

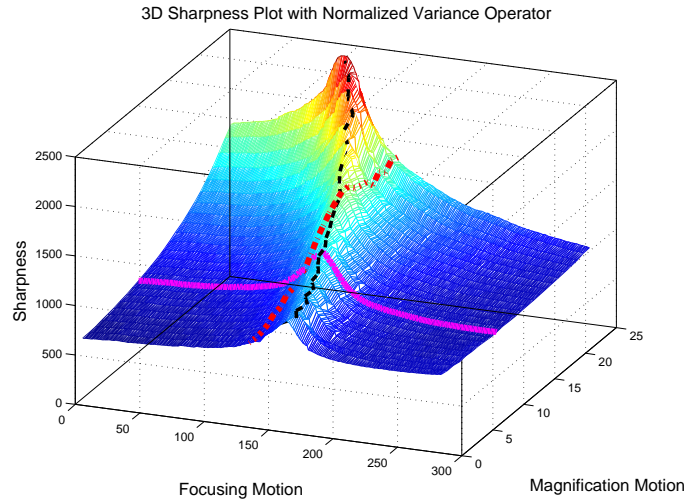


Figure 7.12: Coupled Effect of Magnification and Focus over the Sharpness

The results obtained in the prior experiments are given in Figure 7.12. In order to illustrate the coupling between magnification and focus, this figure include a 3D plot of sharpness with respect to motion in magnification motor and focusing motor above. Besides that 3D plot, two other plots are included in Figure 7.13 that contain the 2D plots obtained via cutting Figure 7.12 along magnification and focus axes respectively. From the figures, it is obvious that there is only a single maximum focus point for each magnification level. Hence, one can draw a curve that represents the maximum focus with respect to a given magnification level. In the figures, those curves are highlighted with dotted black line.

In order to better illustrate the relationship between the maximum sharpness curve and the magnification level, the maximum focus points corresponding to each different magnification level is picked and plotted with respect to the magnification. This plot is given in Figure 7.14 below.

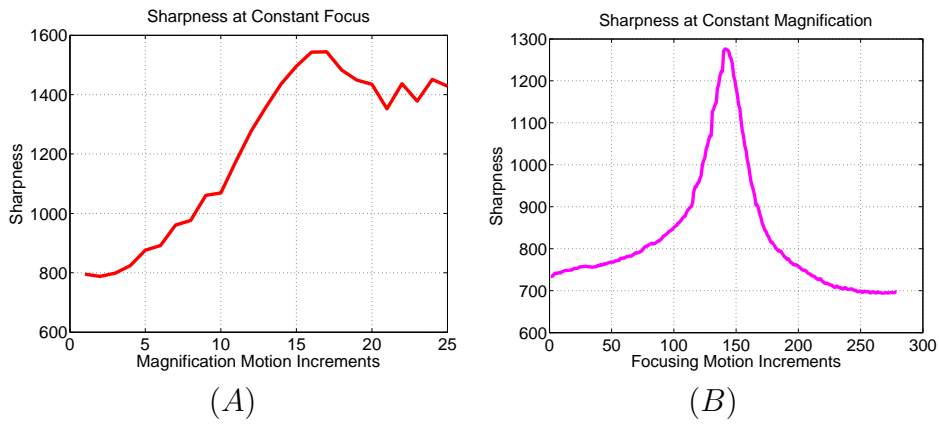


Figure 7.13: Effect of Different Motions over Sharpness; Magnification vs. Sharpness (A), Focusing vs. Sharpness (B)

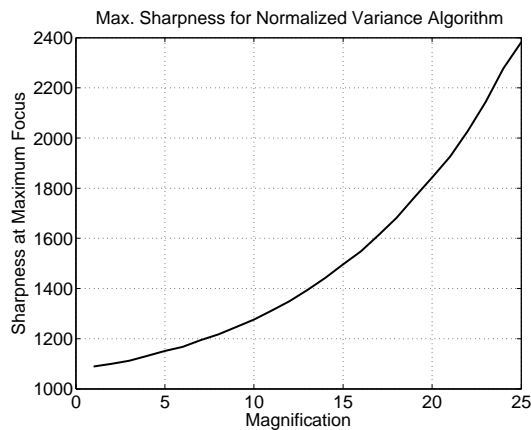


Figure 7.14: Maximum Focus Point Sharpness vs. Magnification Plot

Looking at the behavior of the curve comparing the maximum focus points with respect to a given magnification level (i.e. Figure 7.14), one can observe the fact that the general shape resembles a quadratic rise up of sharpness with respect to increasing magnification. However, such a deterministic approach is almost impossible since the sharpness level is dependent on the image under the scope. In that sense, methods like curve fitting or polynomial

interpolation will only work for a pre-analyzed image which will not make sense for robust tracking of focus point during real time experiments with arbitrary images.

Making use of the feasible information provided by the system through the prior experiments given above, one can conclude that a self optimizing structure can work for focusing the system at a given magnification level. However, self optimizing structures rely strongly on gradient information and require the optimization argument be a unimodal variable. Considering the entire motion range, normalized variance sharpness measure provides a such unimodal shape. Hence, the self-optimization algorithm given below can be realized over the system using Normalized Variance sharpness measuring scheme.

The optimization algorithm described in the next section takes its origins from the scheme presented in [123]. The main idea behind that self-optimization scheme is to have a varying motion reference until the extremum point is caught. The algorithm makes use of boundary layers to guide the motion in such a way that the tracking error can only be zero at the optimum point and is always positive otherwise and the controller is adjusted to have a decreasing error (either exponentially or at finite time). However, unlike the signum function based switching boundaries presented in the original study, here we introduce a well known continuous decision boundary (i.e. the sigmoid function) for smooth reference generation. Moreover, the proposed algorithm does not require any apriori information related to the maximum value of the gradient, as proposed in the original study, in exchange of the requirement of gradient calculation during optimization.

Let $f(x)$ be a convex, continuous and differentiable function with $df(x)/dx \neq$

0 everywhere but in the minimum point x^* and let \dot{x} be the time-derivative (i.e. $dx(t)/dt$) of the optimization function argument. The proposed controller assumes velocity input for the motion control system under scope. Then, under the assumption of ideal behavior, the gradient of the function $f(x)$ is given by a simple derivative as follows;

$$Grad_f = \frac{df(x)}{dx} \quad (156)$$

However, in real applications, there is always a noise component that frequently alters the smooth behavior of function resulting in miscalculations of the gradient. Hence, for the implementation purposes the gradient of the function f is calculated over a window which can be reformulated by;

$$Grad_f \triangleq g_f = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \quad (157)$$

where, we have $x_2 = x_1 + \delta$ with $\delta > 0$. Here, the parameter δ defines the width of the window and should be selected based on the structure of existing noise in the system. Following the definition of gradient, for self optimizing controller, we are proposing the following velocity reference;

$$\dot{x}_{ref} = \mu_0 \left\{ 1 - \frac{2}{1 + e^{-\alpha g_f}} \right\} \quad (158)$$

where, $\mu_0 > 0$ and $\alpha > 0$ are the parameters that determine the amplitude of reference velocity and convergence speed (i.e. extremum vicinity behavior) of the algorithm respectively. Assuming the existence of a velocity tracking structure for the optimization argument, the function given in (158) implies a positive [negative] velocity reference whenever the gradient has negative

[positive] value, hence enforces convergence to the minimum point.

For the system under consideration (i.e. autofocusing with motorized system) optimization argument is the focusing motor position $q(t)$ whereas, the function to be optimized is the Normalized Variance image sharpness which is denoted by $S(q)$. Moreover, since the maximum focus point coincides with the maximum sharpness value, during implementation of the proposed controller, the function to be minimized is selected as $-S(q)$.

For actuators of the plant under consideration, one can make use of the disturbance observer structure as presented in (23). Once the disturbance term is canceled, we now come up with a system that can accept and track acceleration references with an acceleration control framework. The only remaining part is to derive the desired acceleration for the disturbance observer integrated plant. In order to have the acceleration reference for the system under scope, we can start by defining the control error. Since we have the velocity reference from the optimizing controller, one can define the following error;

$$\varepsilon = \dot{q}_{ref} - \dot{q} \quad (159)$$

where, \dot{q}_{ref} has the content defined by equation (158). Due to the convergence requirement, one can impose an exponentially decaying error structure for the tracking of reference velocity as follows;

$$\dot{\varepsilon} + C\varepsilon = 0 \quad (160)$$

with $C > 0$ determining the rate of convergence of error to zero value. Substituting equation (159) to (160), one can extract the following desired ac-

celeration, which concludes the derivation of self optimizing controller;

$$\ddot{q}_{des} = \ddot{q}_{ref} + C\dot{q}_{ref} - C\dot{q} \quad (161)$$

The structure of the system with controller is given in Figure 7.15 while the results obtained via the outlined self optimizing controller scheme are presented in the next chapter.

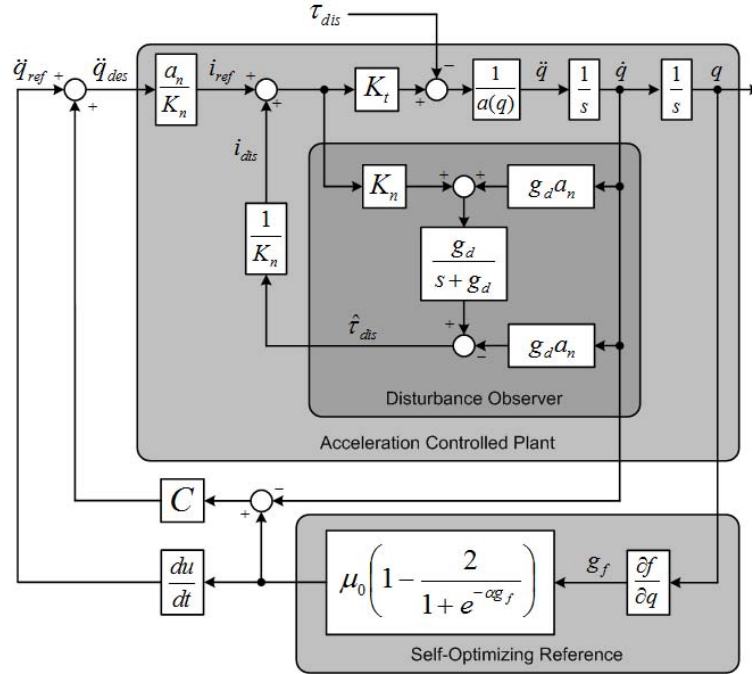


Figure 7.15: Structure of Self Optimizing Controller

7.2.3 A Novel Data Compression Method for Master Slave Communication

The final constituent for the realization of the proposed algorithms is the network environment between the master and slave systems. The data sent

between geographically separated systems is transmitted over the network medium using some transmission protocols like User Datagram Protocol (UDP) or Transmission Control Protocol (TCP). Since TCP is more reliable with less packet losses during data transmission, in the context of this study TCP based communication is preferred between the master and slave systems.

It is well known that the bandwidth of the signal with respect to the sampling period is related by Nyquist theory. However, in networked control systems, this phenomena gets more complicated [124]. The complication arise due to the conflicting nature of frequencies between the controller system and network data transmission. Smaller sampling intervals are required to enforce stability and better precision on the controller side. On the other hand, higher frequency of data packet transmission is undesired in the network side since it amplifies the above mentioned problems [125]. This gives rise to realization of two Nyquist frequencies for the acquisition of a signal; one that is determined by the sampling period for the control, and the other determined by the packet transmission rate [124]. Same type of problems exist in the biomedical area to transmit bio-potential signals over the network (i.e. ECG signals). The solution for the transmission over network is obtained via the use of compression schemes.

Application of compression schemes over networked control systems has recently been popularized by some researchers [126]. Incorporation of a compression algorithm within the control loop provides the opportunity to have high frequency in the control loop while having relatively low sampling frequency over network. In that sense, codec schemes based on DCT [124] and DFT [127] are proposed to be used in bilateral teleoperation systems. A

depiction of teleoperation system with compression is shown below in Figure 7.16.

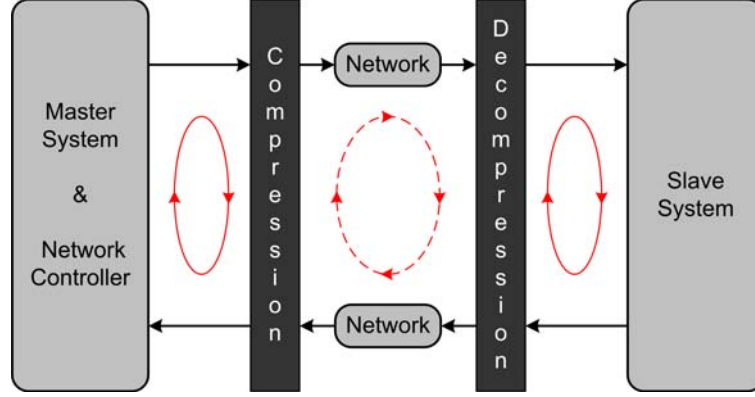


Figure 7.16: Representative Scheme of Teleoperation System with Signal Compression

In the context of this study, a novel scheme for haptic data compression is proposed and used. The proposed scheme is a modified version of DCT based compression scheme for networked motion control applications, which we name as Selected Discrete Cosine Transform (SDCT). The proposed method relies on addition of a selection algorithm based on bubble sorting to choose the coefficients of transformed signal that carry the maximum energy.

DCT based compression structure works on the principle of representing the original signal with coefficients obtained from the so called Discrete Cosine Transform and is intensely used particularly for computer vision applications [128]. In order to obtain the compressed signal, the original signal is passed through the following mathematical transformation;

$$\vec{c}_{DCT}[k] = \sqrt{\frac{2}{N}} C(k) \sum_{n=0}^{N-1} x[n] \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad (162)$$

where, x , $DCT[\cdot]$ and N stand for the main input data, array of DCT coefficients and the length of buffer that keeps the input data respectively. In equation (162), the coefficient C can be obtained from;

$$C(k) = \left\{ \begin{array}{ll} 1/\sqrt{2} & \text{if } k = 0 \\ 1 & \text{if } k \neq 0 \end{array} \right\} \quad (163)$$

Once the original signal is transformed, the high frequency components of the transformed signal are truncated from the buffer and the remaining low frequency components (\vec{c}_{DCT}) are stored in a new coefficient array. The compression of the signal occurs at this stage with a cost that certain amount of data from the original signal is lost. In other words, high frequency components of the original signal are lost in order to achieve coarse reconstruction of original signal by using less amount of data. This procedure is applied over the reference current signal of time delayed motion control system. Following the compression of signal, selected representative coefficients are sent to the other end of the network to generate the necessary control input for tracking of the slave system. A representative schematic of this process is shown below in Figure 7.17.

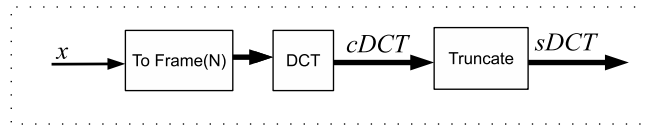


Figure 7.17: Representation of DCT Based Compression Scheme

On the remote side of the overall system, as the compressed signal is received from network, the decompression phase starts. The decompression of signal is handled in a similar way to that of compression by using the Inverse Discrete

Cosine Transform which has the mathematical form given in equation (164).

$$\hat{x}(k) = \sqrt{\frac{2}{N}} C(k) \sum_{n=0}^{N-1} \vec{c}_{DCT}(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad (164)$$

here, \hat{x} , \vec{c}_{DCT} and C stand for the decompressed signal, array of DCT coefficients which carry the information of the original signal and the coefficient vector described before in equation (163). While decompressing the signal, the truncated values (i.e. values corresponding to high frequency components) are taken as zero. The representation of decompression procedure is given in Figure 7.18.

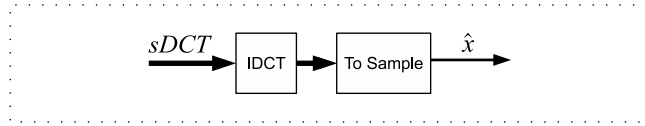


Figure 7.18: Representation of DCT Based Decompression Scheme

In the proposed methodology, unlike the conventional DCT [125], the representative signal coefficients to be transmitted to the remote side is selected using a particular algorithm. Here, instead of selecting the coefficients in the predefined frequency interval, the coefficients that carry the maximum power are select as the signal carriers. With the proposed method, the signal under consideration is transmitted with greater entropy then the traditional method. The algorithmic recipe used for the proposed selected DCT method is given below in Algorithm 4.

Algorithm 4 Selected DCT Based Compression Decompression

```
1: procedure sDCTBASEDCOMPRESSION( $i(n)$ )
2:    $\vec{x} \leftarrow [i(n)i(n-1) \dots i(n-N+1)]$ 
3:   for  $k:=0$  to  $N-1$  do
4:     if  $k = 0$  then
5:        $C = \sqrt{\frac{1}{N}}$ 
6:     else
7:        $C = \sqrt{\frac{2}{N}}$ 
8:     end if
9:      $\vec{c}_{DCT}(k) = C \sum_{n=0}^{N-1} \vec{x}[n] \cos\left(\frac{(2n+1)k\pi}{2N}\right)$ 
10:  end for
11:   $\vec{l} = [1 : \text{length}(\vec{c}_{DCT})]$ 
12:  for  $i:=1$  to  $\text{length}(\vec{c}_{DCT}) - 1$  do
13:    for  $j:=i+1$  to  $\text{length}(\vec{c}_{DCT})$  do
14:      if  $\vec{c}_{DCT}(i) > \vec{c}_{DCT}(j)$  then
15:         $\vec{c}_{DCT}(i) \rightleftharpoons \vec{c}_{DCT}(j)$ 
16:         $\vec{l}(i) \rightleftharpoons \vec{l}(j)$ 
17:      end if
18:    end for
19:  end for
20:  for  $i:=1$  to  $\vec{c}_{DCT}$  do
21:    if  $\vec{l}(i) < m$  then
22:       $\vec{c}_{sDCT}(i) = \vec{c}_{DCT}(i)$ 
23:    else
24:       $\vec{c}_{sDCT}(i) = 0$ 
25:    end if
26:  end for
27: end procedure
28: procedure sDCTBASEDDECOMPRESSION( $\vec{c}_{DCT}$ )
29:   for  $k:=1$  to  $N$  do
30:     if  $k = 0$  then
31:        $C = \sqrt{\frac{1}{N}}$ 
32:     else
33:        $C = \sqrt{\frac{2}{N}}$ 
34:     end if
35:      $\hat{x}[k] = C \sum_{n=0}^{N-1} \vec{c}_{DCT}(n) \cos\left(\frac{(2n+1)k\pi}{2N}\right)$ 
36:   end for
37: end procedure
```

Chapter VIII

8 Experiment Results

The results obtained from various experiments are presented in the following subsections. In that sense, first, results from open contour tracking experiments are illustrated. Tracking responses are shown for two different open contour shapes and laser cutting results over wood are shown as proof of concept. Following that, results acquired from the proposed contouring controller structure are shown and compared with standard independent joint controller structure. The contouring controller experiments are done over two different set of figures both of which prove the success of the proposed methodology. Further, results obtained from assistive machining experiments are presented for remote cutting of two different shapes over fiberglass material under laser. Once the major results obtained from the proposed controllers are presented, the experiment responses obtained from other proposed algorithms are presented as supplementary material to the overall structure of the thesis. In that sense, the proposed kinematic methodology is verified on one of the delta robot setups (i.e. the master setup) first. Following that, the results obtained from self-optimizing controller for autofocusing of inspection unit are presented which conclude the experimental illustration chapter. Below, the details related to these experiments are elab-

orated along with the necessary discussion on salient features of experiment results.

8.1 Open Contour Machining Experiments

In order to have the experimental validation for open contour machining method proposed in Chapter 4, several hand written letters were tested under the slave robot of experimental setup. In that sense, first, the pictures drawn on a white paper are grabbed by the inspection unit for the letters "s" and "e". Following the acquisition of letters, the image processing methodology covered in Chapter 3 is realized to get the corresponding reference trajectories. Once the position, velocity and acceleration references for constant velocity tracking are generated and mapped for the workspace, they are fed to the preview control scheme illustrated in Figure 4.2. The tracking results of experiments for constant 0.002 m/s tangential velocity references for letters "e" and "s" are shown below in Figure 8.1 and Figure 8.2 respectively.

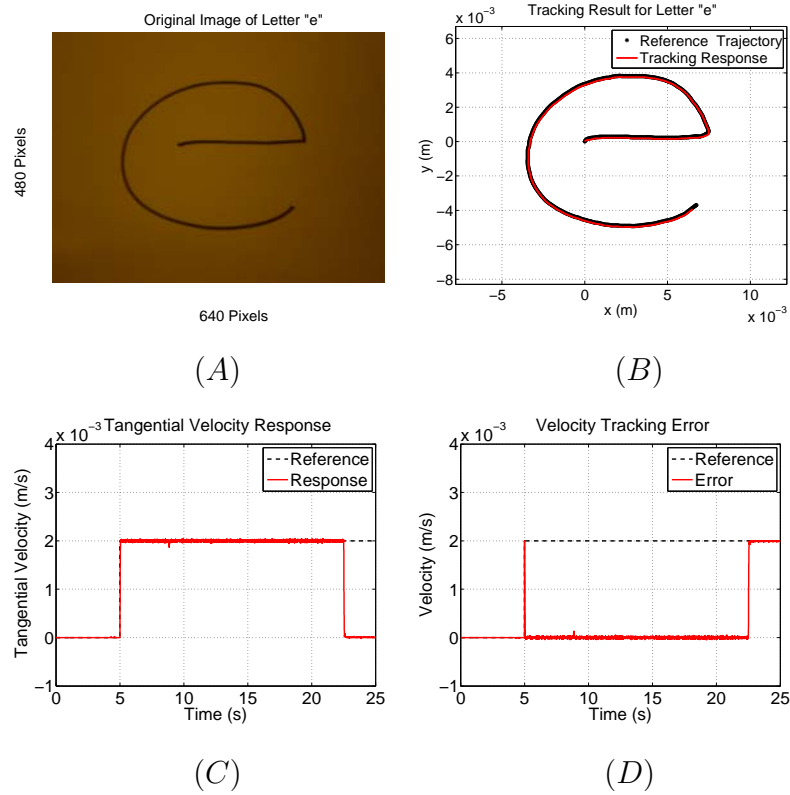


Figure 8.1: Performance of Constant Velocity Tracking Algorithm with Preview Control. Original RGB Image of Letter "e" (A), Tracking Response on X-Y Plane for Letter "e" (B), Tangential Velocity Tracking Response for Letter "e" (C), Tangential Velocity Tracking Error for Letter "e" (D)

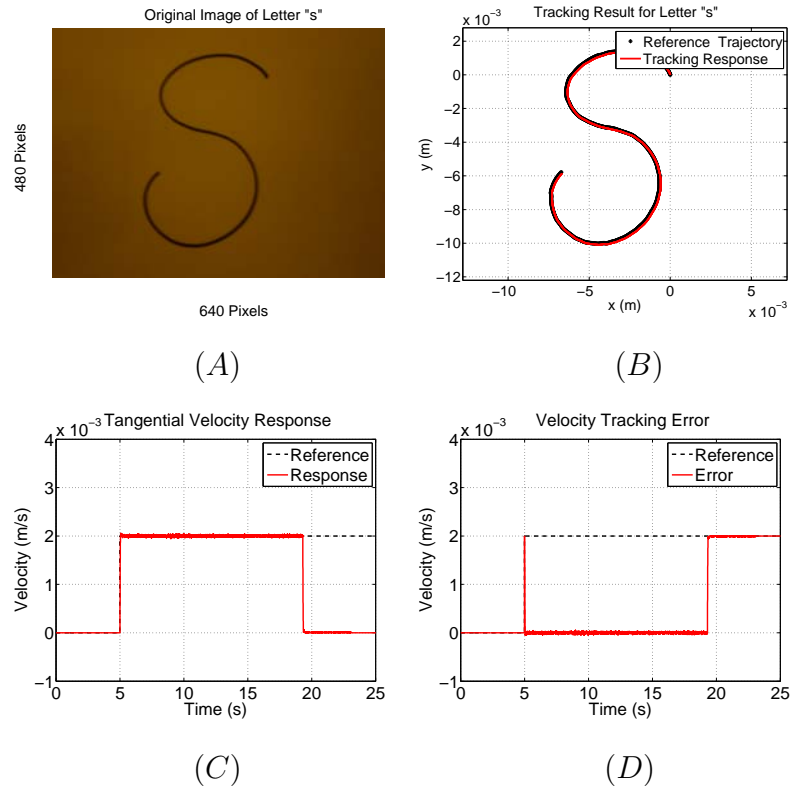


Figure 8.2: Performance of Constant Velocity Tracking Algorithm with Preview Control. Original RGB Image of Letter "s" (A), Tracking Response on X-Y Plane for Letter "s" (B), Tangential Velocity Tracking Response for Letter "s" (C), Tangential Velocity Tracking Error for Letter "s" (D)

In order to further justify the results, responses from laser cutting process with the proposed controller are also shown below in Figure 8.3. The goal in these experiments is to show possibility of laser marking on an incident surface with certain desired open contour shape rather than cutting. Here, it is important to note that due to the interaction between laser beam and wood and due to lack of long focal length, the cutting depth of wood can only be limited to some value around $1mm$. This is because of the lack of long

focal length optical components on the laser being used during experiments. The selection of tangential velocity as $2mm/s$ relies on this fact.

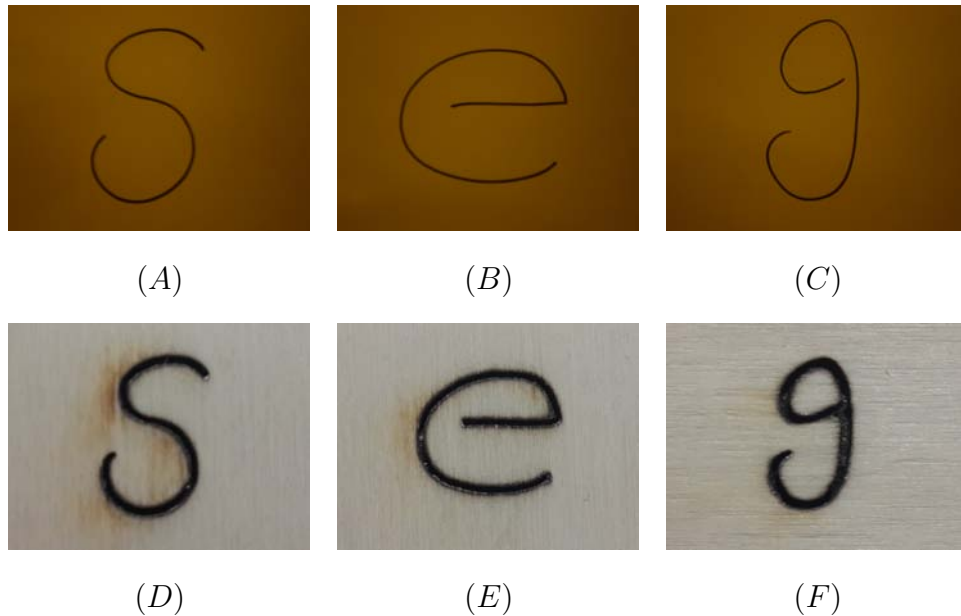


Figure 8.3: Results from Laser Cutting Experiments for Open Contour Shapes. Original RGB Images (A), (B) (C), Response of Laser Cutting on Wood (D), (E), (F)

As obvious from the results, the overall preview controller algorithm can perform good for smooth curves while there are deviations from the constant velocity reference at points where corners exist (i.e. right corner of letter "e" in 9th second of Figure 8.1 C). These deviations at sharp corners would have less influence for smaller tangential velocity references. This way, the bounds of the algorithm in maintaining the reference for sharp corners can also be determined as part of future study.

8.2 Closed Contour Machining Experiments

Experimental verification of closed contour machining process with the proposed contouring control algorithm is illustrated in this subsection. In order to give a wider perspective on the applicability of the proposed method, two different sets of experiments are carried out. In the first set, images of CAD drawings for certain shapes made on a computer program is used as the input for the overall image processing and contouring control algorithm and position tracking responses are illustrated. In the second experiment set, images of various shapes encountered in real life application scenarios and corresponding laser cutting responses over various materials are preferred to further validate the proposed scheme.

8.2.1 Contouring Control Experiment Set-1

In this experiment set, three images from CAD generated images are used. The contents of these images contain an eight leg machine cam, a gear and an octagon as shown in Figure 8.4 (A), (B) and (C) respectively. The 2D trajectories constructed from EFD models of these curves are shown below the original images in Figure 8.4 (D), (E) and (F). In these figures, the tracking responses with standard independent joint control structure and with the proposed contour tracking controller are also shown. In order to have a better judgement, the zoomed plots of these figures are also shown below in Figure 8.4 (G), (H) and (I) respectively. Finally, the performance of proposed orthogonal axis based contour tracking controller is further depicted via the plots of tracking errors in normal direction in Figure 8.4 (J), (K) and (L). It should be pointed out that for these geometries, reference velocity in tangential direction is selected as $8mm/s$.

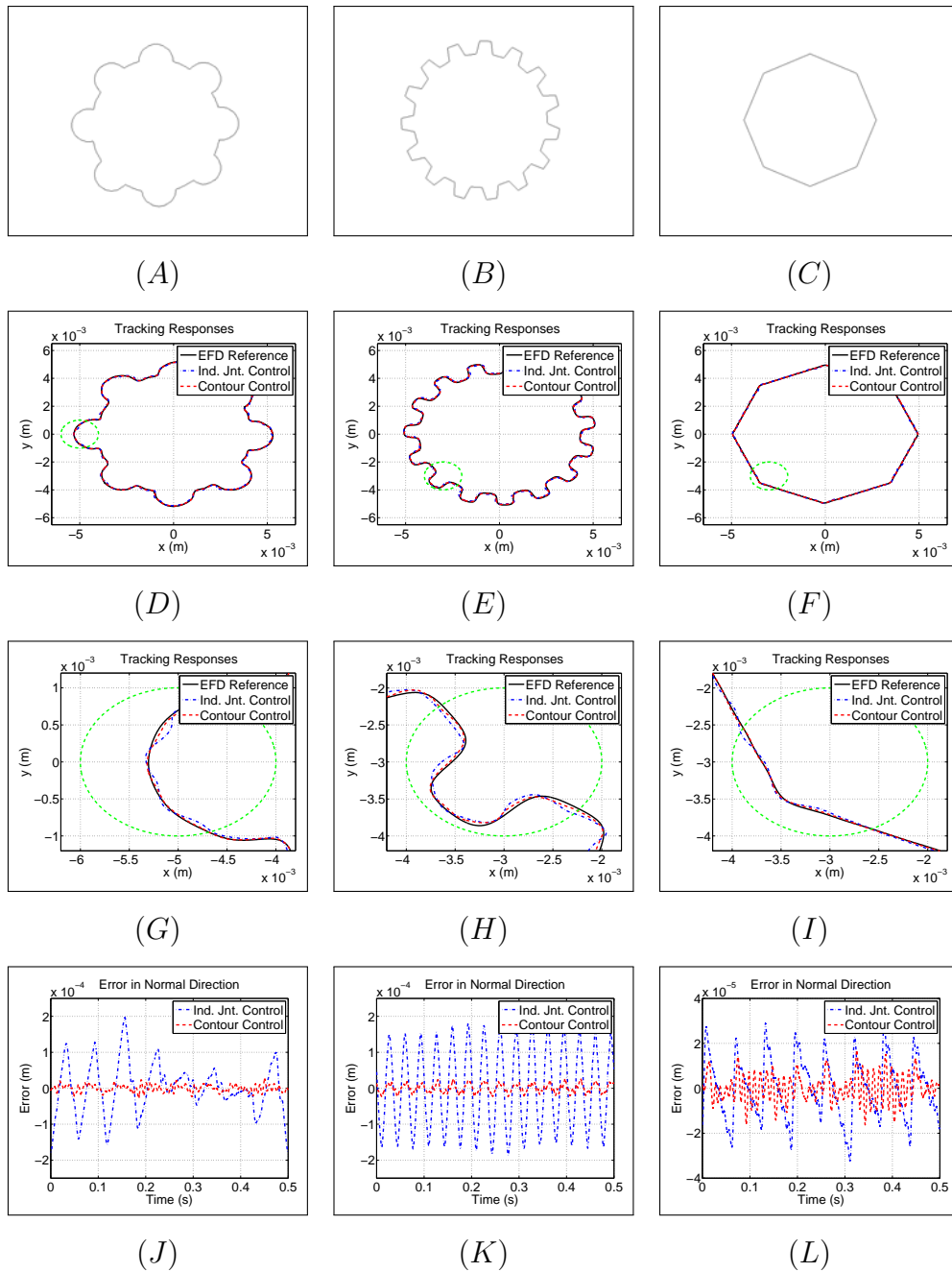


Figure 8.4: Performance of Contouring Control Algorithm. Images of CAD Generated Shapes (A), (B), (C); EFD Models and Tracking Responses (D), (E), (F); Zoomed Plots (G), (H), (I); Normal Direction Tracking Errors (J), (K), (L)

8.2.2 Contouring Control Experiment Set-2

In this experiment set, figures of shapes from random images are selected as input to the overall algorithm. In order to also show the potential application of the proposed framework over medical applications like laser surgery. In that sense, image from a melanoma region over skin, image of a cancerous tissue on throat and image of a random shape drawn over a paper are shown in Figure 8.5 (A), (B) and (C) respectively. The detected contours and EFD models fitted over these contours are correspondingly shown in Figure 8.5 (D), (E) and (F). Further, laser cutting responses of the first two shapes and laser marking of the third shape made on thin black colored fiberglass material and wood are respectively shown in Figure 8.5 (G), (H) and (I). Finally in Figure 8.5 (J), (K) and (L) the velocity tracking responses in tangent direction for machining of these three figures are given.

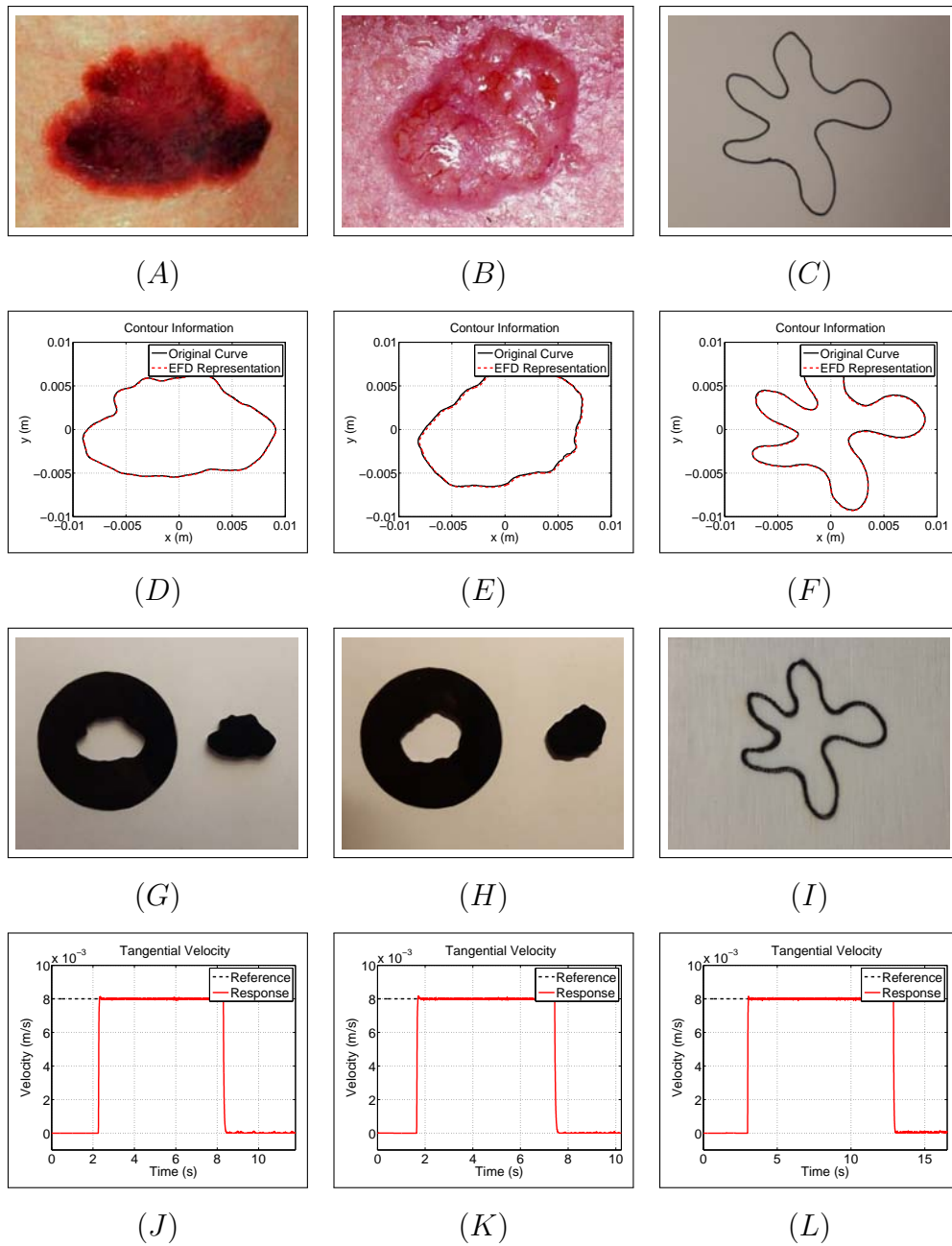


Figure 8.5: Performance of Contouring Control Algorithm. Images of Real Shapes (A), (B), (C); Detected Contours and Fitted EFD Curves (D), (E), (F); Laser Cutting Response on Fiberglass (G), (H); Laser Marking Response on Wood (I); Tangent Direction Contouring Velocities (J), (K), (L)

From the given figures, the success and wide range application potential of contouring control is obvious. Especially in high speed and high precision machining applications, contouring controller outperforms the standard independent joint control structure by providing much less error in normal direction. This is basically due to directly acting over normal direction error using the orthogonal coordinate system over the reference contour.

8.3 Constrained Manual Machining Experiments

Making use of the platform described above, experiments for the proposed constrained remote laser machining framework are carried out. In that sense, a human operator used the master robot of the experimental platform and conducted random motion over the confined space of this robot. In other words, the master robot's end effector motion is constrained with the reference contour and the human operator carried out free motion over this virtual constraint. For better illustration purposes, assistive manual laser cutting and marking of two different reference shapes are made again over fiberglass and wood respectively.

The content of contours for constrained remote laser machining experiments are extracted from the images given in Figure 8.6 (A) and (B) respectively. Both of the figures that show the experiment results contain information about master robot cartesian position (A), slave robot cartesian position (B), master robot motion and re-scaled slave robot motion with β^{-1} on the same graph (C), master robot tangent direction velocity response (D), slave robot tangent direction velocity response (E) and tangential velocities of master and re-scaled slave robots on the same graph (F). For the best illustration purpose, the laser cutting results over fiberglass are depicted in (G) and (H)

and laser marking results over wood is depicted in (I). In the first experiment (Figure 8.7), the scaling factor is taken as % 50 of the original master motion trajectory for reference slave motion (i.e. $\beta = 0.5 \times \mathbf{I}_3$). On the other hand, the second experiment (Figure 8.8) is done with a scaling factor of 2/3 (i.e. $\beta = 0.666 \times \mathbf{I}_3$). One important point about these results is that the traced trajectory does not show one to one corresponding with the marked region since marking is made only while the operator is pushing the laser firing button of the pen over master system. Another important issue is that the tangential velocity is not anymore constant in these results. That is basically due to the random and free motion of human operator over the constrained contour. Results are given in Figure 8.7 and Figure 8.8 below.

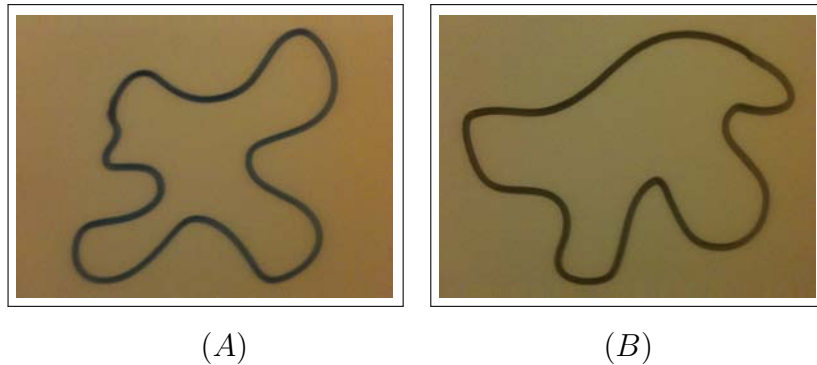


Figure 8.6: Original RGB Images for Constraining Master Robot Motion. Experiment-1 (A), Experiment-2 (B)

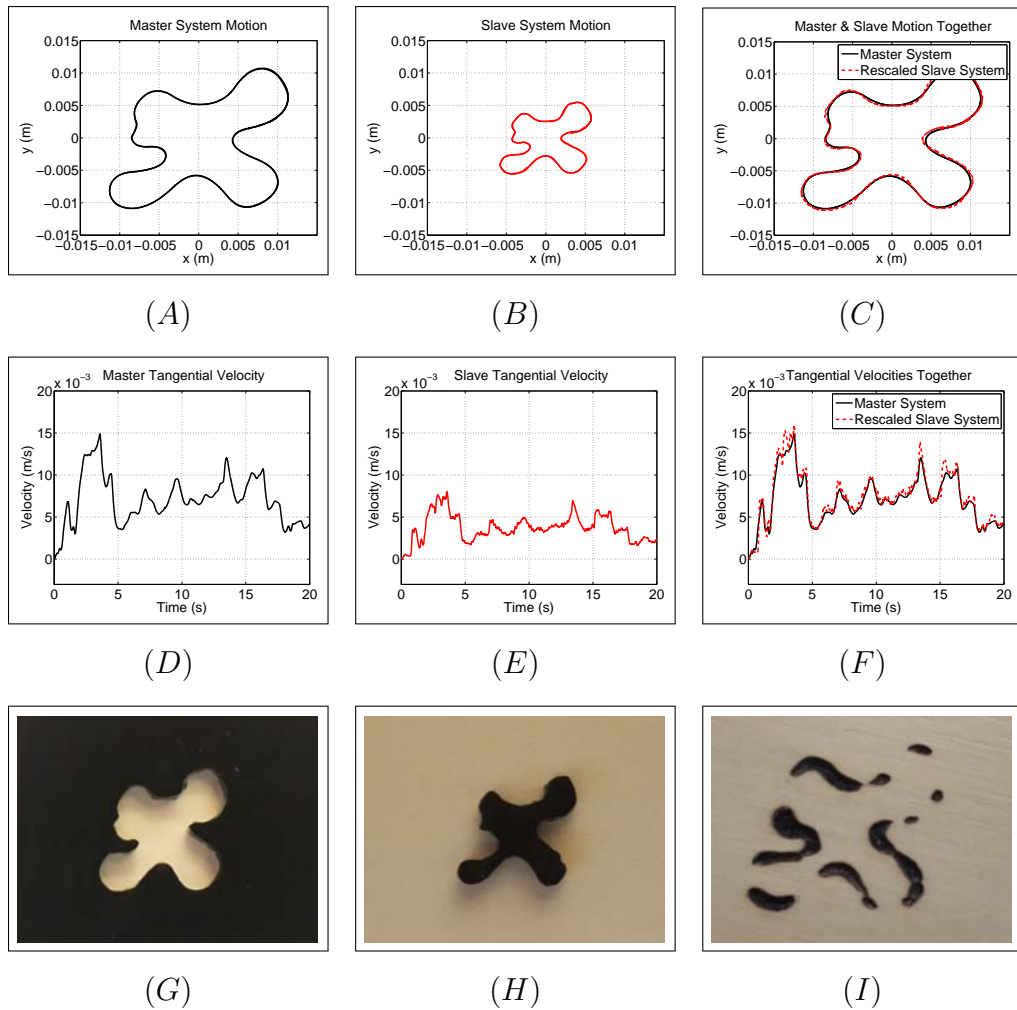


Figure 8.7: Constrained Remote Laser Machining Experiment 1. Master Motion (A), Slave Motion (B), Master and Re-Scaled Slave Motion (C), Tangential Velocity Responses (D), (E), (F); Fiberglass Cutting Response (G), (H); Wood Marking Response (I)

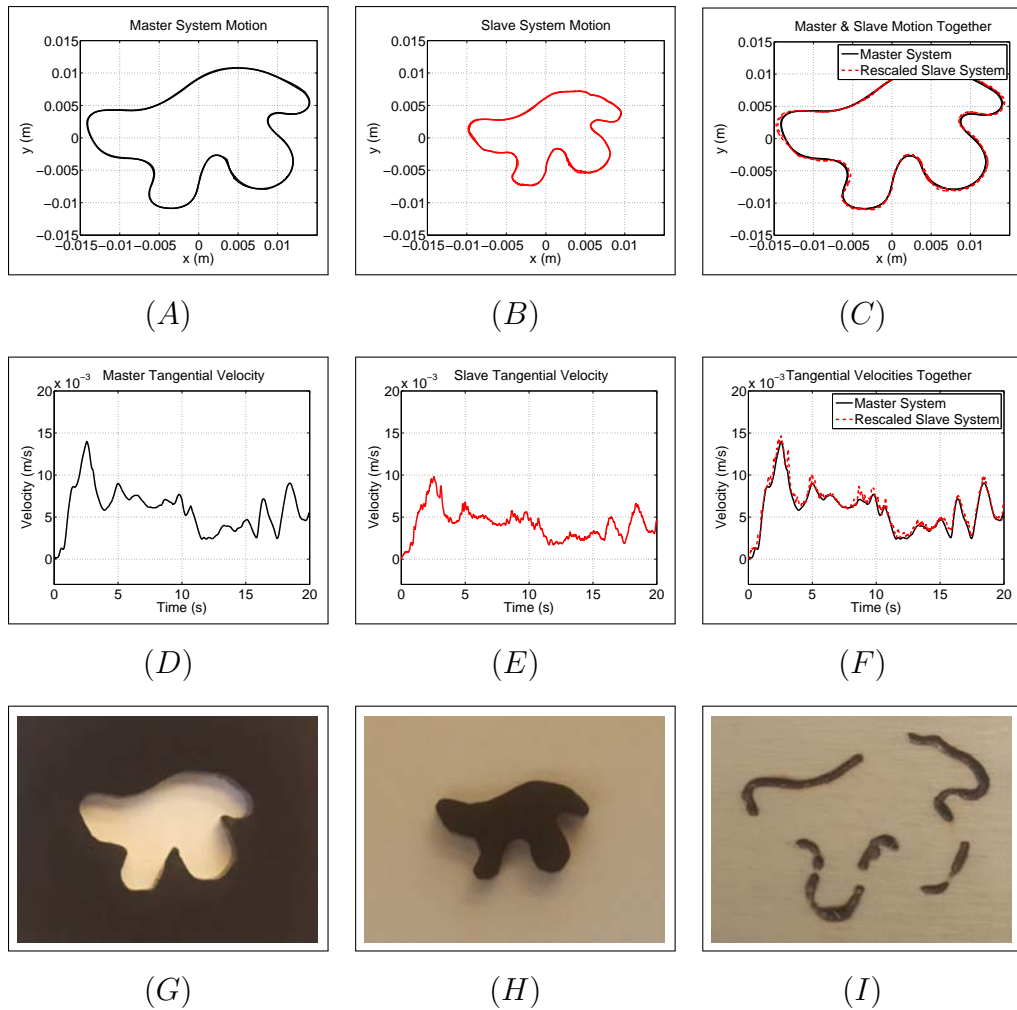


Figure 8.8: Constrained Remote Laser Machining Experiment 2. Master Motion (A), Slave Motion (B), Master and Re-Scaled Slave Motion (C), Tangential Velocity Responses (D), (E), (F); Fiberglass Cutting Response (G), (H); Wood Marking Response (I)

Results obtained from these experiments show the success of the proposed framework on constrained remote non-contact machining via a master & slave robotic platform. With this achievement, the possibility of robot assisted laser surgery from a distant location can be made feasible opening new

paradigms in this direction. Potential extensions of this study is elaborated in the future work section of Chapter 9.

8.4 Generalized Kinematics Validation Experiments

In order to validate the kinematics derivation made in Chapter 6, another 3-DOF manipulator is designed and attached to the end effector of the delta robot. This second robot is structurally a serial cartesian manipulator consisting of linear actuators driven by rotary motors and ball-screw mechanisms. The motors are from PI-M111 series equipped with internal encoders that can sense as small as $50nm$ of linear motion due to high gain in ball screw. During the experiments this 3-DOF cartesian manipulator is actuated with step references in all of x , y and z axes and the motion of the end effector is recorded along with the real-time measurements of delta robot joint positions and real-time calculations of forward and inverse kinematics. The results obtained from the experiments are explained in the following subsection. A depiction of the experimental platform is given below in Fig. 8.9.

For the validation purposes, a set of experiments is made in the setup described above. In the experiments, the cartesian manipulator is controlled under step references of $5mm$ and the real-time responses are measured. For the verification of forward kinematics, the position response of the cartesian manipulator is compared with the calculated forward kinematics response of delta robot using the joint displacements and the results are shown below in Figure 8.10 (A), (B) and (C) for end effector x , y and z motions respectively. In the figures, the abbreviations FK and TS stand for forward kinematics and task space respectively.

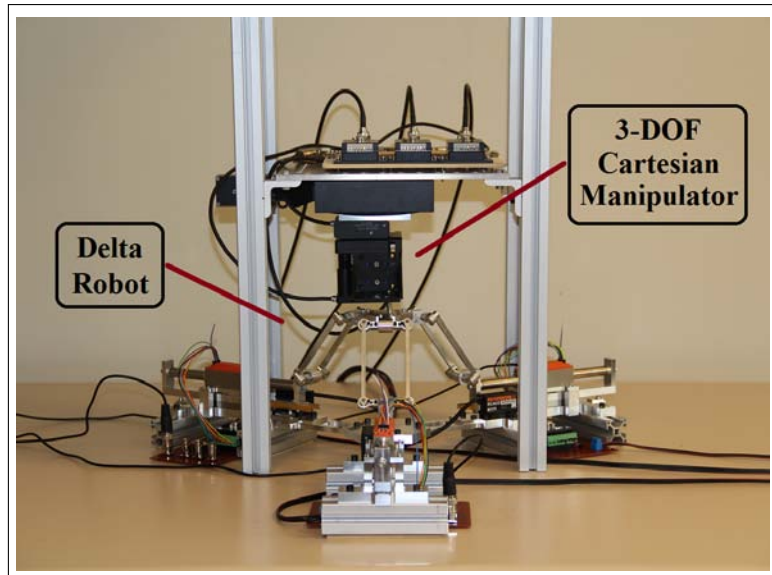


Figure 8.9: Experiment Platform For Verification of Generalized Kinematics

Inverse kinematics is also verified in a similar way to that of forward kinematics. For that purpose, the motion of the cartesian manipulator is used as the input for inverse kinematics algorithm and the responses are compared with the measurements obtained from the joints of delta robot. The results of inverse kinematics validation are illustrated in Figure 8.10 (*D*), (*E*) and (*F*) below. Likewise, the abbreviations *IK* and *JS* represent the inverse kinematics and joint space respectively.

Verification of the kinematic jacobian is made following a little more complicated path. The position measurements of the joints of delta robot are differentiated to get the joint velocities. These velocities are then multiplied by the kinematic jacobian to calculate the end effector velocities. These calculated velocities are compared with the velocities obtained by differentiating the position measurement of cartesian manipulator and the results are shown below in Figure 8.10 (*G*), (*H*) and (*I*). In these figures, the data with label

”Measured” represent the differentiated end effector position while the data with label ”Calculated” stand for the velocities obtained from joints using jacobian.

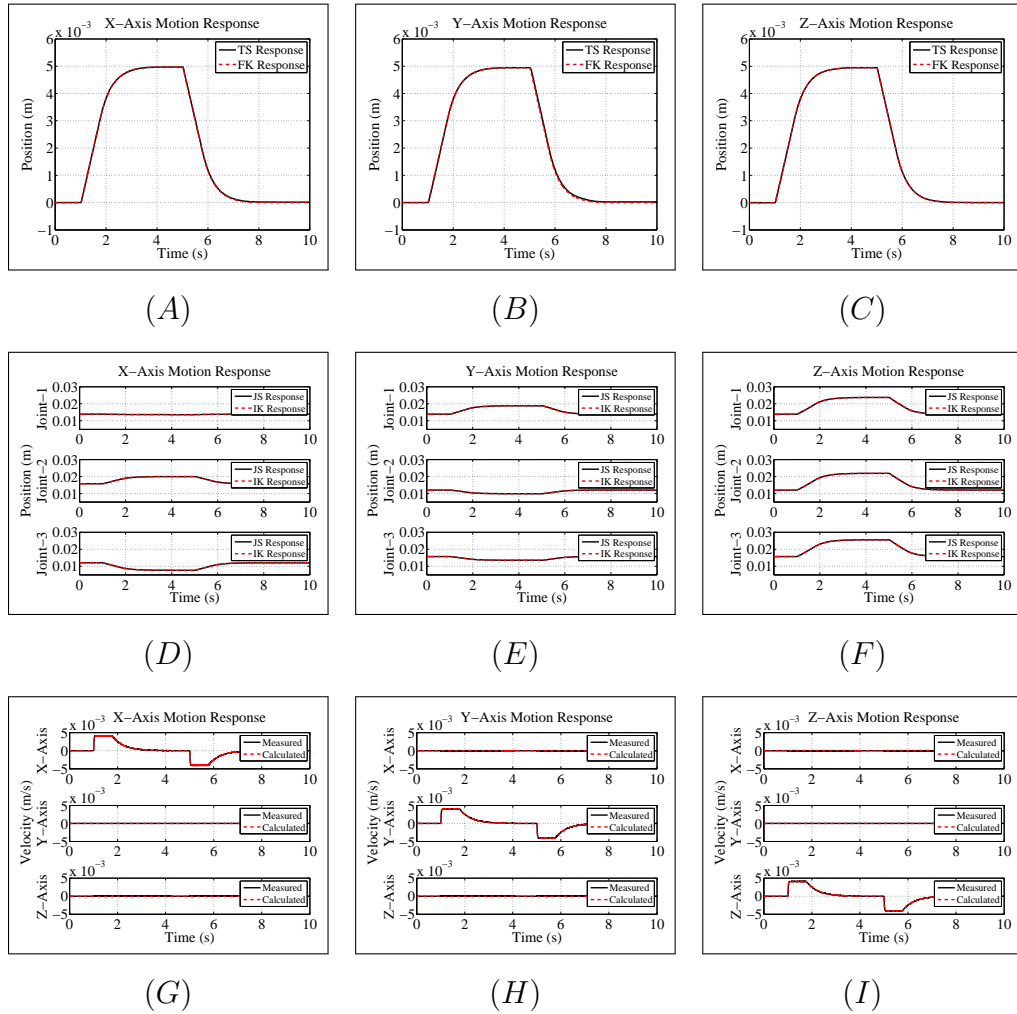


Figure 8.10: Generalized Kinematics Algorithm Experiment Responses. Verification of Forward Kinematics (A), (B), (C); Verification of Inverse Kinematics (D), (E), (F); Verification of Kinematic Jacobian (G), (H), (I);

As obvious from the figures, the formulation made throughout Chapter 6

shows one to one correspondence with the actual measurements, concluding the proof of concept part of the study.

8.5 Autofocusing Experiments

The proposed self optimizing controller structure is implemented on the setup described in Chapter 7. In order to verify the optimization algorithm, experiments are carried out for testing the robustness of the system against changes in the magnification level. In that sense, two different experiments are made. In the first experiment, a defocused system is taken and the magnification is increased from its current level. The algorithm automatically moves the focusing motor until the maximum sharpness value is achieved. In the second experiment, a computer generated reference is given to the magnification motor to zoom out on an apriori focused image. With the changes in the zoom level, the system is pushed to a defocused region while the focusing algorithm is running. The response of the algorithm is observed thereafter until it recovers back the maximum sharpness point at that particular magnification. The responses from these experiments are shown below in Figure8.11 and Figure8.12 respectively.

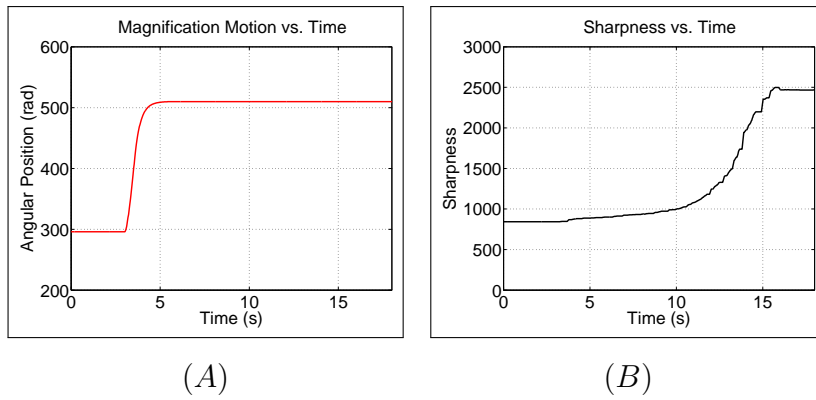
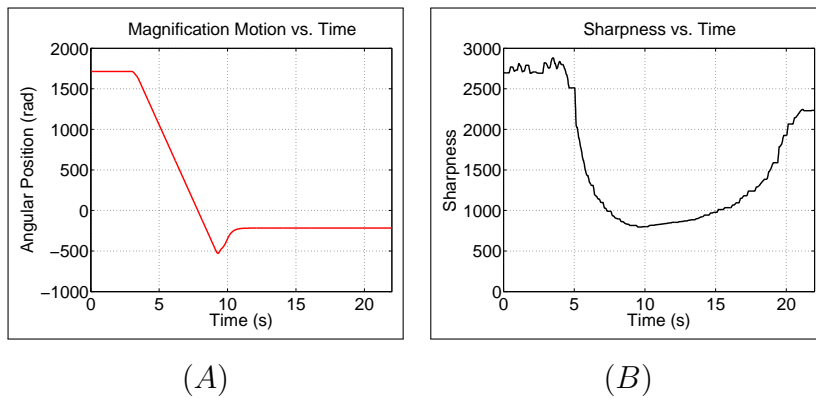


Figure 8.11: Autofocusing Experiment-1. Change in Magnification Level (A), Change in Sharpness Level (B)



(C)

Figure 8.12: Autofocusing Experiment-2. Change in Magnification Level (A), Change in Sharpness Level (B), Process of Manual Defocusing and Automatic Focusing (C)

In both of the experiments, the first data shows the changes in the magnification level whereas the second data shows how the sharpness of system

changes while the algorithm traces the maximum focus point. For the second experiment, the images taken during this manual defocusing and automatic focusing procedure is also given below the plots of sharpness and magnification to provide further insight about the overall process. As obvious from the sharpness plots of the system and from the recorded pictures, the algorithm is robust against the disturbance created by the magnification motor and can track the gradient until the maximum focus point for the current level of magnification is recovered. It should be noted here that the effects of noise in the measurement is compensated via using a relatively large window for gradient during implementation. Better results would be expected if any advanced method (like polynomial interpolation) is tried to further smooth the image gradient. Moreover, since the algorithm works based on gradient (i.e. change of sharpness), when there is no excitation in the system, it does not start working. The algorithm starts to work once magnification is changed and continuously traces the maximum focus point thereafter.

Chapter IX

9 Conclusion & Future Work

9.1 Conclusion

In this thesis, a motion control framework for non-contact cutting of random shaped contours is presented for applications both in industrial and medical processes. In that context, first, the problem of contour information generation is handled. An easy to use, methodological way based on a combination of several image processing techniques is proposed for the acquisition of reference geometry to be machined. Based on desired application, the geometry obtained from a single image is used to generate either time dependent cartesian trajectories or time independent cartesian constraint set.

In order to process the generated reference trajectory during non-contact cutting operation by a motion control system, two different schemes are utilized based on robust motion control framework. In the first scheme, a slight modification of independent joint control, namely the preview control, is adopted to handle the constant tangential velocity tracking requirement of shapes with sharp edges. In the second scheme, benefitting from the possibility of modeling a closed curve with Elliptic Fourier Descriptors, a new control methodology is proposed. This new structure relies on definition of a new

coordinate system that moves with the reference operational space contour and that is always orthogonal to the motion direction. Defining the error and formulating the controller in this new space, the tracking precision is further enhanced enabling the application of proposed framework for high speed and high precision laser cutting as required in industrial applications.

Proceeding further, the non-contact machining process is extended for application in a master slave platform yielding towards assistive applications in medical surgeries. In that sense, making use of the constraints generated from the reference contour, a method is proposed to confine the master operator motion over a virtual contour that has the same shape of original trajectory. In the proposed method, the human operator can move freely along the contour direction and make cutting of certain segments over that contour. Enforcing a slave system track the reference motion exhibited by the human operator on the master system, the proposed method extends the non-contact cutting ability to a distant location. This way, high precision remote non-contact cutting of certain profiles is made possible opening a new door for remote laser surgery operations.

For the realization of proposed remote non-contact machining scheme, direct drive prismatically actuated delta robots are preferred in both master and slave systems. In order to enable a wider range of realization, a generalized kinematics formulation for prismatically actuated 3 DOF parallel robots is presented that is valid for all configurational variations including the asymmetric distribution of actuators. Further, experiment specific algorithms and methods are proposed for the realization of the overall framework.

Experimental validation for all of the proposed and adopted schemes is made in a platform that contains an image acquisition unit, two structurally differ-

ent delta robots and a laser machining unit. The results obtained from the experiments prove the validity of the overall framework for both industrial and medical applications.

9.2 Future Work

The proposed framework in the context of this thesis opens new paradigms in many different directions. In that sense, some of the potential extensions for the proposed methods can be listed as follows;

- *Adaptive Preview Control*: The open contour machining structure adopted by the preview control is prone to imperfections at sharp edges due to rapid acceleration or deceleration. In order to overcome this problem, a velocity adaptation structure can be combined with the preview control methodology and laser machining unit. This way, the controller can adjust the cutting depth much better at sharp edges which improves the overall machining quality.
- *3D Contouring Control*: The contouring control presented in this thesis benefits from the orthogonal moving coordinate system over the reference trajectory and performs much better than independent joint control scheme. Extending the same idea of moving orthogonal coordinate frame for tracking of trajectories in 3D would also contribute to classical machining processes like CNC machining. Results acquired in this direction will be much valued directly in industrial applications.
- *Non-Contact Haptic Feedback*: Considering the formulation presented under the constrained systems baseline, possibility of making a new formulation to obtain force feedback from a non-contact machining

system comes into picture. In that sense, following a precise modeling between the remote machining unit (i.e. the laser cutting system) and the material being processed, a virtual copy of the machined surface could be generated on the master side of a overall remote machining system. Using a similar methodology, the motion of master system could be constrained to trace the surface shape. Successful achievement in that direction will lead the possibility of virtual force feedback on the master side enabling the human operator feel the machined surface much better. Such an achievement will find important use in remote laser surgery systems.

10 Appendices

10.1 Appendix A: Realization of Constraints

Realization of the controller and methodology is done on a master/slave structure two robot system each of which consists of 3-DOF prismaticly actuated parallel delta type robots. The details related to the experimental platform is presented in the experiments section. In order to realize the controller derived in the previous section, correct definition of the constraints should be made. Since in the approach described in Section 3.3 the constraint on task space is defined as a combination of circle segments, the prior requirement from the system is the satisfaction of this constraint. Moreover, since the system is supposed to keep the z -axis of the operational space at a constant value, another constraint could be inserted over the system. This way, the constraint vector can be formulated as;

$$\phi(\mathbf{x}) = \begin{bmatrix} x^2 + y^2 - r^2 \\ z^{ref} - z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (165)$$

where, task space coordinates x , y and z are all functions of time and z^{ref} is assumed to take some constant value. The radius r of circle is determined using the algorithm described in section 3.3. With this definition of constraints, the jacobian matrix associated with the constraints can be given as;

$$\mathbf{J}_\phi = \begin{bmatrix} 2x & 2y & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (166)$$

10.2 Appendix B: Proof of invertibility of Matrix A

Matrix A of equation (114) consists of terms given in equation (110). The invertibility of this matrix can be verified by checking its determinant. However, direct substitution of σ_{ij} will generate a highly complicated expression for the determinant. Rather than that we can first simplify the content of matrix A and then analyze further. For this purpose, one can define the following intermediate expression;

$$\zeta_i = d_i \cos(\alpha_i) + r_e - r_b \quad (167)$$

With that expression in hand, substituting the content of abbreviated σ_{ij} terms of (110) back to (114), one can write the content of matrix A as follows;

$$A = 2 \begin{bmatrix} c\theta_1\zeta_1 - c\theta_3\zeta_3 & s\theta_1\zeta_1 - s\theta_3\zeta_3 \\ c\theta_2\zeta_2 - c\theta_3\zeta_3 & s\theta_2\zeta_2 - s\theta_3\zeta_3 \end{bmatrix} \quad (168)$$

where, $c\theta_i$ and $s\theta_i$ again stand for the cosine and sine terms respectively. Now, with that structure of matrix A , one can calculate its determinant as;

$$\begin{aligned} \det(A) &= -\zeta_1\zeta_2c\theta_2s\theta_1 + \zeta_1\zeta_2c\theta_1s\theta_2 \\ &\quad -\zeta_1\zeta_3c\theta_1s\theta_3 + \zeta_1\zeta_3c\theta_3s\theta_1 \\ &\quad -\zeta_2\zeta_3c\theta_3s\theta_2 + \zeta_2\zeta_3c\theta_2s\theta_3 \end{aligned} \quad (169)$$

which can be further simplified using trigonometric identities as follows;

$$\begin{aligned}
\det(A) &= \zeta_1 \zeta_3 \sin(\theta_1 - \theta_3) \\
&\quad - \zeta_1 \zeta_2 \sin(\theta_1 - \theta_2) - \zeta_2 \zeta_3 \sin(\theta_2 - \theta_3)
\end{aligned} \tag{170}$$

Looking at the system geometry given in Figure 6.1, one can observe that the terms $\zeta_i < 0$ and hence $\zeta_i \zeta_j > 0$ for all $i, j \in \{1, 2, 3\}$. Having kept this in mind, the analysis of equation (170) can proceed with the analysis of sine terms. Referring to the angle definitions given in Figure 6.1, one can write down the following information related to the signs of sine terms;

$$\begin{aligned}
-2\pi < (\theta_1 - \theta_3) < -\pi &\Rightarrow \sin(\theta_1 - \theta_3) > 0 \\
-\pi < (\theta_1 - \theta_2) < 0 &\Rightarrow \sin(\theta_1 - \theta_2) < 0 \\
-\pi < (\theta_2 - \theta_3) < 0 &\Rightarrow \sin(\theta_2 - \theta_3) < 0
\end{aligned} \tag{171}$$

With these information related to the signs of the terms in (170), one can conclude that for all feasible values of angles θ_i and all values $\alpha_i \in (0, \pi/2)$ with $i \in 1, 2, 3$, we have;

$$\det(A) > 0 \tag{172}$$

■

References

- [1] A. Hace and K. Jezernik, “Control system for the waterjet cutting machine,” *Mechatronics, IEEE/ASME Transactions on*, vol. 9, no. 4, pp. 627–635, 2004.
- [2] Q. Guo, J. Li, and X. Dai, “Cutting quality prediction of a quasi-5-axis abrasive waterjet machine with an adjustable workhead,” in *Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on*. IEEE, 2012, pp. 181–186.
- [3] D. Kalyanasundaram, “Mechanics guided design of hybrid laser/waterjet system for machining of hard and brittle materials,” 2009.
- [4] D. R. Quintero, W. Mejia, and J. Rosero, “Good practice for electric discharge machining (edm) bearing currents measurement in the induction motor and drives system,” in *Electric Machines & Drives Conference (IEMDC), 2013 IEEE International*. IEEE, 2013, pp. 1384–1390.
- [5] W. Mysinski, “Power supply unit for an electric discharge machine,” in *Power Electronics and Motion Control Conference, 2008. EPE-PEMC 2008. 13th*. IEEE, 2008, pp. 1321–1325.

- [6] A. Hace, K. Jezernik, and M. Terbuc, "Vss motion control for a laser-cutting machine," *Control Engineering Practice*, vol. 9, no. 1, pp. 67–77, 2001.
- [7] C. Hnatovsky, R. Taylor, E. Simova, P. Rajeev, D. Rayner, V. Bhardwaj, and P. Corkum, "Fabrication of microchannels in glass using focused femtosecond laser radiation and selective chemical etching," *Applied Physics A*, vol. 84, no. 1-2, pp. 47–61, 2006.
- [8] D. Lee, "Modeling of high speed remote laser cutting of electrodes for lithium-ion batteries," Ph.D. dissertation, The University of Michigan, 2012.
- [9] J. Diaci, D. Bračun, A. Gorkič, and J. Možina, "Rapid and flexible laser marking and engraving of tilted and curved surfaces," *Optics and lasers in engineering*, vol. 49, no. 2, pp. 195–199, 2011.
- [10] J. Burgner, M. Mueller, J. Raczkowski, and H. Woern, "Robot assisted laser bone processing: Marking and cutting experiments," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*. IEEE, 2009, pp. 1–6.
- [11] L. Lu, B. Yao, and W. Lin, "A two-loop contour tracking control for biaxial servo systems with constraints and uncertainties," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 6468–6473.
- [12] H.-C. Ho, J.-Y. Yen, and S.-S. Lu, "A decoupled path-following control algorithm based upon the decomposed trajectory error," *International Journal of Machine Tools and Manufacture*, vol. 39, no. 10, pp. 1619–1630, 1999.

- [13] P. Ouyang, V. Pano, and T. Dam, “Pid position domain control for contour tracking,” *International Journal of Systems Science*, no. ahead-of-print, pp. 1–14, 2013.
- [14] H. Meng, Y. Lou, and J. Chen, “High speed contouring control of biaxial systems based on task polar coordinate frame,” in *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on*. IEEE, 2013, pp. 1583–1588.
- [15] J. Yang, D. Zhang, and Z. Li, “Position loop-based cross-coupled control for high-speed machining,” in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*. IEEE, 2008, pp. 4285–4290.
- [16] G.-C. Chiu and M. Tomizuka, “Contouring control of machine tool feed drive systems: a task coordinate frame approach,” *Control Systems Technology, IEEE Transactions on*, vol. 9, no. 1, pp. 130–139, 2001.
- [17] L. Tang and R. Landers, “Multiaxis contour control the state of the art,” *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 6, pp. 1997–2010, 2013.
- [18] X. Wang, N. Liu, and M. Wang, “Research and implementation of high-precision biaxial tracking control system based on nurbs interpolator,” *The International Journal of Advanced Manufacturing Technology*, vol. 52, no. 1-4, pp. 255–262, 2011.
- [19] S. Abrate and D. Walton, “Machining of composite materials. part ii: Non-traditional methods,” *Composites Manufacturing*, vol. 3, no. 2, pp. 85–94, 1992.

- [20] W. Wei, Z. Di, D. Allen, and H. Almond, “Non-traditional machining techniques for fabricating metal aerospace filters,” *Chinese Journal of Aeronautics*, vol. 21, no. 5, pp. 441–447, 2008.
- [21] J. Meijer, “Laser beam machining (lbm), state of the art and new opportunities,” *Journal of Materials Processing Technology*, vol. 149, no. 1, pp. 2–17, 2004.
- [22] Y. Qin, A. Brockett, Y. Ma, A. Razali, J. Zhao, C. Harrison, W. Pan, X. Dai, and D. Loziak, “Micro-manufacturing: research, technology outcomes and development issues,” *The International Journal of Advanced Manufacturing Technology*, vol. 47, no. 9-12, pp. 821–837, 2010.
- [23] H. Becker and C. Gärtner, “Polymer microfabrication methods for microfluidic analytical applications,” *Electrophoresis*, vol. 21, no. 1, pp. 12–26, 2000.
- [24] S. Qin and W. J. Li, “Process characterization of fabricating 3d micro channel systems by laser-micromachining,” *Sensors and Actuators A: Physical*, vol. 97, pp. 749–757, 2002.
- [25] V. Maselli, R. Osellame, G. Cerullo, R. Ramponi, P. Laporta, L. Magagnin, and P. L. Cavallotti, “Fabrication of long microchannels with circular cross section using astigmatically shaped femtosecond laser pulses and chemical etching,” *Applied physics letters*, vol. 88, no. 19, p. 191107, 2006.
- [26] Q. Sun, A. Saliminia, F. Théberge, R. Vallée, and S. L. Chin, “Microchannel fabrication in silica glass by femtosecond laser pulses with

- different central wavelengths,” *Journal of Micromechanics and Micro-engineering*, vol. 18, no. 3, p. 035039, 2008.
- [27] N. Bloembergen, “Laser-material interactions; fundamentals and applications,” in *Laser ablation: mechanisms and applications II*, vol. 288, no. 1. AIP Publishing, 1993, pp. 3–10.
- [28] J. Miller, “History, scope, and the future of laser ablation,” in *Laser Ablation*. Springer, 1994, pp. 1–10.
- [29] Y. Kawamura, K. Toyoda, and S. Namba, “Effective deep ultraviolet photoetching of polymethyl methacrylate by an excimer laser,” *Applied Physics Letters*, vol. 40, no. 5, pp. 374–375, 1982.
- [30] R. Srinivasan and V. Mayne-Banton, “Self-developing photoetching of poly (ethylene terephthalate) films by far-ultraviolet excimer laser radiation,” *Applied Physics Letters*, vol. 41, no. 6, pp. 576–578, 1982.
- [31] F. Dabby and U.-C. Paek, “High-intensity laser-induced vaporization and explosion of solid material,” *Quantum Electronics, IEEE Journal of*, vol. 8, no. 2, pp. 106–111, 1972.
- [32] M. Modest and H. Abakians, “Heat conduction in a moving semi-infinite solid subjected to pulsed laser irradiation,” *Journal of heat transfer*, vol. 108, no. 3, pp. 597–601, 1986.
- [33] R. Berenyi, “Simulating the laser micromachining of a 3d flexible structure,” *Microsystem technologies*, vol. 15, no. 12, pp. 1855–1860, 2009.
- [34] G. Chryssolouris, *Laser machining*. Springer, 1991.

- [35] J. Li, D. Qin, H. Li, and Y. Chen, “Research on high speed laser carving control system based on embedded os,” in *Mechatronics and Automation, 2009. ICMA 2009. International Conference on.* IEEE, 2009, pp. 2188–2192.
- [36] A. Ancona, F. Röser, K. Rademaker, J. Limpert, S. Nolte, and A. Tünnermann, “High speed laser drilling of metals using a high repetition rate, high average power ultrafast fiber cpa system,” *Optics express*, vol. 16, no. 12, pp. 8958–8968, 2008.
- [37] A. K. Dubey and V. Yadava, “Laser beam machining a review,” *International Journal of Machine Tools and Manufacture*, vol. 48, no. 6, pp. 609–628, 2008.
- [38] P. Parandoush and A. Hossain, “A review on modelling and simulation of laser beam machining,” *International Journal of Machine Tools and Manufacture*, 2014.
- [39] G. Chryssolouris, P. Sheng, and F. von Alvensleben, “Process control of laser grooving using acoustic sensing,” *Journal of Manufacturing Science and Engineering*, vol. 113, no. 3, pp. 268–275, 1991.
- [40] R. Lausten and P. Balling, “On-the-fly depth profiling during ablation with ultrashort laser pulses: A tool for accurate micromachining and laser surgery,” *Applied Physics Letters*, vol. 79, no. 6, pp. 884–886, 2001.
- [41] T. Inoue, M. Morimoto, and K. Ohnishi, “A preview controller with time based spline approximation for multi-axis manipulator,” *IEEE Industrial Electronics, IECON*, pp. 247–251, 2006.

- [42] R. G. Landers, "Error space motion control methodology for complex contours," *Asian Journal of Control*, vol. 7, no. 1, pp. 20–28, 2005.
- [43] M. Tomizuka, "Zero phase error tracking algorithm for digital control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 1, pp. 65–68, 1987.
- [44] H. Zhang and R. G. Landers, "Precision motion control methodology for complex contours," *Journal of Manufacturing Science and Engineering*, vol. 129, no. 6, pp. 1060–1068, 2007.
- [45] A. Visioli, G. Ziliani, and G. Legnani, "Friction compensation in hybrid force/velocity control for contour tracking tasks," *Industrial Robotics: Theory, Modeling and Control*, pp. 875–894, 2006.
- [46] K. Srinivasan and P. Kulkarni, "Cross-coupled control of biaxial feed drive servomechanisms," *Journal of dynamic systems, measurement, and control*, vol. 112, no. 2, pp. 225–232, 1990.
- [47] S.-S. Yeh and P.-L. Hsu, "Estimation of the contouring error vector for the cross-coupled control design," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 1, pp. 44–51, 2002.
- [48] L. Wang, F. Jin, and Y. Sun, "Contour control for direct drive xy table," in *Mechatronics and Automation, 2009. ICMA 2009. International Conference on*. IEEE, 2009, pp. 4919–4923.
- [49] A. El Khalick M and N. Uchiyama, "Contouring controller design based on iterative contour error estimation for three-dimensional machining," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 4, pp. 802–807, 2011.

- [50] J. R. Conway, C. A. Ernesto, R. T. Farouki, and M. Zhang, “Performance analysis of cross-coupled controllers for cnc machines based upon precise real-time contour error measurement,” *International Journal of Machine Tools and Manufacture*, vol. 52, no. 1, pp. 30–39, 2012.
- [51] S.-L. Chen and K.-C. Wu, “Contouring control of smooth paths for multiaxis motion systems based on equivalent errors,” *Control Systems Technology, IEEE Transactions on*, vol. 15, no. 6, pp. 1151–1158, 2007.
- [52] Y. Koren, “Cross-coupled biaxial computer control for manufacturing systems,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 4, pp. 265–272, 1980.
- [53] Y. Koren and C. Lo, “Advanced controllers for feed drives,” *CIRP Annals-Manufacturing Technology*, vol. 41, no. 2, pp. 689–698, 1992.
- [54] F. Huo and A.-N. Poo, “Improving contouring accuracy by using generalized cross-coupled control,” *International Journal of Machine Tools and Manufacture*, vol. 63, pp. 49–57, 2012.
- [55] G.-J. Wang and T.-J. Lee, “Neural-network cross-coupled control system with application on circular tracking of linear motor xy table,” in *Neural Networks, 1999. IJCNN’99. International Joint Conference on*, vol. 3. IEEE, 1999, pp. 2194–2199.
- [56] M.-T. Yan, M.-H. Lee, and P.-L. Yen, “Theory and application of a combined self-tuning adaptive control and cross-coupling control in a retrofit milling machine,” *Mechatronics*, vol. 15, no. 2, pp. 193–211, 2005.

- [57] P. Kulkarni and K. Srinivasan, "Optimal contouring control of multi-axial feed drive servomechanisms," *ASME Journal of Engineering for Industry*, vol. 111, no. 2, pp. 140–148, 1989.
- [58] K. Barton, J. van de Wijdeven, A. Alleyne, O. Bosgra, and M. Steinbuch, "Norm optimal cross-coupled iterative learning control," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 3020–3025.
- [59] Y. Wu, J. Shi, C. Su, and Q. Zou, "A control approach to cross-coupling compensation of piezotube scanners in tapping-mode atomic force microscope imaging," *Review of Scientific Instruments*, vol. 80, no. 4, p. 043709, 2009.
- [60] J.-H. Chin, Y.-M. Cheng, and J.-H. Lin, "Improving contour accuracy by fuzzy-logic enhanced cross-coupled precompensation method," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 1, pp. 65–76, 2004.
- [61] L. Tang and R. G. Landers, "Predictive contour control with adaptive feed rate," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 4, pp. 669–679, 2012.
- [62] S.-L. Chen, H.-L. Liu, and S. C. Ting, "Contouring control of biaxial systems based on polar coordinates," *Mechatronics, IEEE/ASME Transactions on*, vol. 7, no. 3, pp. 329–345, 2002.
- [63] B. Yao, C. Hu, and Q. Wang, "An orthogonal global task coordinate frame for contouring control of biaxial systems," *Mechatronics, IEEE/ASME Transactions on*, vol. 17, no. 4, pp. 622–634, 2012.

- [64] L. Tang and R. Landers, “Multiaxis contour control the state of the art,” *Control Systems Technology, IEEE Transactions on*, vol. 21, no. 6, pp. 1997–2010, 2013.
- [65] R. Hecker, G. Flores, Q. Xie, and R. Haran, “Servocontrol of machine tools: a review,” *Latin American applied research*, vol. 38, no. 1, p. 85, 2008.
- [66] A. Trevisani, A. Rossi, and V. Zanotto, “Design and implementation of a mechatronic device for robot-assisted neurosurgery,” in *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*. IEEE, 2005, pp. 7–12.
- [67] G. Ho, W. S. Ng, M. Y. Teo, C.-K. Kwoh, and W. S. Cheng, “Computer-assisted transurethral laser resection of the prostate (calrp): theoretical and experimental motion plan,” *Biomedical Engineering, IEEE Transactions on*, vol. 48, no. 10, pp. 1125–1133, 2001.
- [68] H. Monnich, D. Stein, J. Raczowsky, and H. Worn, “An automatic and complete self-calibration method for robotic guided laser ablation,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1086–1087.
- [69] R. Cernat, C. Matei, L. Olteanu, C. Riviere, and D. Dumitras, “Acousto-optic compensation of tremor for use in a handheld laser microsurgical instrument,” in *Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE*. IEEE, 2006, pp. 3862–3864.

- [70] T. Buzung, P. Hering, J. Bongartz, and M. Ivanenko, “A novel navigation principle in computer-assisted surgery,” in *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, 2004, pp. 3132–3135.
- [71] J. W. Motkoski, F. W. Yang, S. H. Lwu, and G. R. Sutherland, “Toward robot-assisted neurosurgical lasers,” *Biomedical Engineering, IEEE Transactions on*, vol. 60, no. 4, pp. 892–898, 2013.
- [72] V. Françoise, A. Sahbani, and G. Morel, “A comanipulation device for orthopedic surgery that generates geometrical constraints with real-time registration on moving bones,” in *Robotics and Biomimetics (RO-BIO), 2011 IEEE International Conference on*. IEEE, 2011, pp. 38–43.
- [73] L. S. Mattos, G. Dagnino, G. Becattini, M. Dellepiane, and D. G. Caldwell, “A virtual scalpel system for computer-assisted laser microsurgery,” in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1359–1365.
- [74] D. Kim, H. Owada, N. Hata, and T. Dohi, “An er: Yag laser bone cutting manipulator for precise rotational acetabular osteotomy,” in *Engineering in Medicine and Biology Society, 2004. IEMBS’04. 26th Annual International Conference of the IEEE*, vol. 1. IEEE, 2004, pp. 2750–2753.
- [75] A. A. Goldenberg, J. Trachtenberg, Y. Yi, R. Weersink, M. S. Sussman, M. Haider, L. Ma, and W. Kucharczyk, “Robot-assisted mri guided prostatic interventions,” *Robotica*, vol. 28, no. 02, pp. 215–234, 2010.

- [76] L. S. Mattos, D. G. Caldwell, M. Dellepiane, and E. Grant, “Design and control of a robotic system for assistive laser phonomicrosurgery,” in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 5411–5415.
- [77] H. Monnich, H. Worn, and D. Stein, “Op sense a robotic research platform for telemanipulated and automatic computer assisted surgery,” in *Advanced Motion Control (AMC), 2012 12th IEEE International Workshop on*. IEEE, 2012, pp. 1–6.
- [78] S. Omori, R. Nakumura, Y. Muragaki, and H. Iseki, “Robotic laser surgery system with volume mapping in neurosurgery,” in *Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference on*. IEEE, 2007, pp. 52–55.
- [79] S. Stopp, D. Svejdar, E. von Kienlin, H. Deppe, and T. C. Lueth, “A new approach for creating defined geometries by navigated laser ablation based on volumetric 3-d data,” *Biomedical Engineering, IEEE Transactions on*, vol. 55, no. 7, pp. 1872–1880, 2008.
- [80] M. Remacle, F. Hassan, D. Cohen, G. Lawson, and M. Delos, “New computer-guided scanner for improving co2 laser-assisted microincision,” *European Archives of Oto-Rhino-Laryngology and Head & Neck*, vol. 262, no. 2, pp. 113–119, 2005.
- [81] K. Harada, M. G. Fujie, T. Nakamura, and T. Chiba, “Bending laser manipulator for intrauterine surgery,” in *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*. IEEE, 2006, pp. 238–242.

- [82] M. M. Desai, M. Aron, I. S. Gill, G. Pascal-Haber, O. Ukimura, J. H. Kaouk, G. Stahler, F. Barbagli, C. Carlson, and F. Moll, “Flexible robotic retrograde renoscopy: description of novel robotic device and preliminary laboratory experience,” *Urology*, vol. 72, no. 1, pp. 42–46, 2008.
- [83] P. R. Rizun and G. R. Sutherland, “Surgical laser augmented with haptic feedback and visible trajectory.” in *VR*, 2005, pp. 241–244.
- [84] P. Rizun and G. Sutherland, “Tactile feedback laser system with applications to robotic surgery,” in *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint.* IEEE, 2005, pp. 426–431.
- [85] L. Fichera, D. Pardo, and L. S. Mattos, “Artificial cognitive supervision during robot-assisted laser surgery.”
- [86] R. Cernat, C. Matei, L. Olteanu, C. Riviere, and D. Dumitras, “Acousto-optic compensation of tremor for use in a handheld laser microsurgical instrument,” in *Engineering in Medicine and Biology Society, 2006. EMBS’06. 28th Annual International Conference of the IEEE.* IEEE, 2006, pp. 3862–3864.
- [87] W. T. Ang, P. Pradeep, and C. Riviere, “Active tremor compensation in microsurgery,” in *Engineering in Medicine and Biology Society, 2004. IEMBS’04. 26th Annual International Conference of the IEEE*, vol. 1. IEEE, 2004, pp. 2738–2741.

- [88] B. C. Becker, C. R. Valdivieso, J. Biswas, L. Lobes, and C. N. Riviere, "Active guidance for laser retinal surgery with a handheld instrument," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*. IEEE, 2009, pp. 5587–5590.
- [89] B. C. Becker, R. A. MacLachlan, L. A. Lobes, and C. N. Riviere, "Semiautomated intraocular laser surgery using handheld instruments," *Lasers in surgery and medicine*, vol. 42, no. 3, pp. 264–273, 2010.
- [90] S. S. Baker, J. M. Hunnewell, W. S. Muenzler, and G. J. Hunter, "Laser blepharoplasty: diamond laser scalpel compared to the free beam co2 laser," *Dermatologic surgery*, vol. 28, no. 2, pp. 127–131, 2002.
- [91] L. S. Mattos, G. Dagnino, G. Becattini, M. Dellepiane, and D. G. Caldwell, "A virtual scalpel system for computer-assisted laser microsurgery," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 1359–1365.
- [92] T. Sielhorst, M. Feuerstein, and N. Navab, "Advanced medical displays: A literature review of augmented reality," *Display Technology, Journal of*, vol. 4, no. 4, pp. 451–467, 2008.
- [93] B. Preising, T. Hsia, and B. Mittelstadt, "A literature review: robots in medicine," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 10, no. 2, pp. 13–22, 1991.
- [94] G. Dagnino, L. S. Mattos, G. Becattini, M. Dellepiane, and D. G. Caldwell, "Comparative evaluation of user interfaces for robot-assisted laser

- phonomicrosurgery,” in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 7376–7379.
- [95] L. S. Mattos and D. G. Caldwell, “Safe teleoperation based on flexible intraoperative planning for robot-assisted laser microsurgery,” in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. IEEE, 2012, pp. 174–178.
- [96] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
- [97] B. D. Lucas, T. Kanade *et al.*, “An iterative image registration technique with an application to stereo vision.” in *IJCAI*, vol. 81, 1981, pp. 674–679.
- [98] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [99] X. Bresson, S. Esedoá,lu, P. Vandergheynst, J.-P. Thiran, and S. Osher, “Fast global minimization of the active contour/snake model,” *Journal of Mathematical Imaging and vision*, vol. 28, no. 2, pp. 151–167, 2007.
- [100] E. A. Baran, E. Golubovic, T. E. Kurt, and A. Sabanovic, “Constant velocity control of a miniature pantograph with image based trajectory generation,” in *Control Conference (ASCC), 2013 9th Asian*. IEEE, 2013, pp. 1–6.

- [101] F. P. Kuhl and C. R. Giardina, "Elliptic fourier features of a closed contour," *Computer graphics and image processing*, vol. 18, no. 3, pp. 236–258, 1982.
- [102] C.-S. Lin and C.-L. Hwang, "New forms of shape invariants from elliptic fourier descriptors," *Pattern recognition*, vol. 20, no. 5, pp. 535–545, 1987.
- [103] Y. H. Esin and M. Ünel, "Formation control of nonholonomic mobile robots using implicit polynomials and elliptic fourier descriptors," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 18, no. 5, pp. 765–780, 2010.
- [104] M.-F. Wu and H.-T. Sheu, "Representation of 3d surfaces by two-variable fourier descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 858–863, 1998.
- [105] J. Miyata, T. Murakami, and K. Ohnishi, "Trajectory tracking control of mobile robot by time based spline approach," *IEEJ Transactions on Industry Applications*, vol. 123, pp. 778–783, 2004.
- [106] J. Miyata and T. Murakami, "Trajectory tracking control of mobile robot by fluid model," *IEEJ Transactions on Industry Applications*, vol. 127, pp. 315–321, 2007.
- [107] L. S. Mattos, D. G. Caldwell, M. Dellepiane, and E. Grant, "Design and control of a robotic system for assistive laser phonomicrosurgery," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 5411–5415.

- [108] J. Yang and Z. Li, “A novel contour error estimation for position loop-based cross-coupled control,” *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 4, pp. 643–655, 2011.
- [109] J. Collins and P. Gremaud, “A simple model for laser drilling,” *Mathematics and Computers in Simulation*, vol. 81, no. 8, pp. 1541–1552, 2011.
- [110] Z. Ji-zhuang and Z. Xue-xue, “Dynamic modeling of tissue ablation with continuous wave co2 laser,” in *Bioinformatics and Biomedical Engineering, 2007. ICBBE 2007. The 1st International Conference on*. IEEE, 2007, pp. 1057–1060.
- [111] K. Ohnishi, M. Shibata, and T. Murakami, “Motion control for advanced mechatronics,” *Mechatronics, IEEE/ASME Transactions on*, vol. 1, no. 1, pp. 56–67, 1996.
- [112] C. O. Saglam, E. A. Baran, A. O. Nergiz, and A. Sabanovic, “Model following control with discrete time smc for time-delayed bilateral control systems,” in *Mechatronics (ICM), 2011 IEEE International Conference on*. IEEE, 2011, pp. 997–1002.
- [113] N. Birla and A. Swarup, “Pso approach to preview tracking control systems,” in *TENCON 2009-2009 IEEE Region 10 Conference*. IEEE, 2009, pp. 1–6.
- [114] A. Casavola and M. Sorbara, “Towards constrained teleoperation for safe long-distance robotic surgical operations,” in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 685–690.

- [115] A. Sabanovic, K. Ohnishi, D. Yashiro, and N. Sabanovic, "Motion control systems with network delay," in *Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE*. IEEE, 2009, pp. 2277–2282.
- [116] Y. Li and Q. Xu, "Kinematic analysis of a 3-prs parallel manipulator," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 4, pp. 395–408, 2007.
- [117] S. Stan, M. Manic, V. Maties, and R. Balan, "Evolutionary approach to optimal design of 3 dof translation exoskeleton and medical parallel robots," in *Human System Interactions, 2008 Conference on*. IEEE, 2008, pp. 720–725.
- [118] Z. Wang, G. Wang, S. Ji, Y. Wan, and Q. Yuan, "Optimal design of a linear delta robot for the prescribed cuboid dexterous workspace," in *Robotics and Biomimetics, 2007. ROBIO 2007. IEEE International Conference on*. IEEE, 2007, pp. 2183–2188.
- [119] M. Bouri and R. Clavel, "The linear delta: Developments and applications," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*. VDE, 2010, pp. 1–8.
- [120] M. Franc and A. Haze, "Fpga implementation of the bilateral control algorithm for a high performance haptic teleoperation," in *Advanced Motion Control (AMC), 2012 12th IEEE International Workshop on*. IEEE, 2012, pp. 1–6.

- [121] Y. Sun, S. Duthaler, and B. J. Nelson, “Autofocusing algorithm selection in computer microscopy,” in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 70–76.
- [122] A. Santos, C. Ortiz de Solorzano, J. J. Vaquero, J. Pena, N. Malpica, and F. Del Pozo, “Evaluation of autofocus functions in molecular cytogenetic analysis,” *Journal of microscopy*, vol. 188, no. 3, pp. 264–272, 1997.
- [123] V. I. Utkin, *Sliding modes in control and optimization*. Springer, 1992, vol. 116.
- [124] M. Mizuochi and K. Ohnishi, “Coding and decoding scheme for wide-band bilateral teleoperation,” in *Advanced Motion Control (AMC), 2012 12th IEEE International Workshop on*. IEEE, 2012, pp. 1–6.
- [125] H. Tanaka, K. Ohnishi, and H. Nishi, “Implementation of lossy haptic data compression using integer dct to fpga,” in *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*. IEEE, 2010, pp. 1726–1731.
- [126] D. Yashiro and K. Ohnishi, “Performance analysis of bilateral control system with communication bandwidth constraint,” *Industrial Electronics, IEEE Transactions on*, vol. 58, no. 2, pp. 436–443, 2011.
- [127] Y. Yokokura, S. Katsura, and K. Ohishi, “Bilateral control using compressor/decompressor under the low-rate communication network,” in *Mechatronics, 2009. ICM 2009. IEEE International Conference on*. IEEE, 2009, pp. 1–6.

- [128] M. S. Moellenhoff and M. W. Maier, “Dct transform coding of stereo images for multimedia applications,” *Industrial Electronics, IEEE Transactions on*, vol. 45, no. 1, pp. 38–43, 1998.

VITA

Abdurrahman Eray Baran
Candidate for the Degree of
Philosophy of Doctorate

Thesis: Assistive Control for Non-Contact Machining of Random Shaped Contours

Major Field: Mechatronics Engineering

Biographical:

Abdurrahman Eray Baran was born in Diyarbakir, Turkey. He received his B.Sc. and M.Sc. degrees in Mechatronics Engineering from Sabanci University, Istanbul, Turkey in 2008 and 2010 respectively. His research interests include motion control systems, task constrained multi-body systems, time delayed systems, bilateral control systems, estimation and observers.

The following papers include results published/submitted out of this thesis with author's primary concessionaire:

- E. A. Baran, A. Kuzu, S. Bogosyan, M. Gokasan, A. Sabanovic, "Comparative Analysis of Selected DCT Based Compression Scheme for Haptic Data Transmission", *Submitted to IEEE Transactions on Industrial Informatics* (SCI)
- E. A. Baran, D. Bilgili, O. Ozen, A. Sabanovic, "Direct Geometric Solution Based Generalized Kinematics of Prismaticly Actuated Parallel Delta Robots for Regular and Uneven Configurations", *Submitted to IEEE Transactions on Mechatronics* (SCI)
- E. A. Baran, Z. Zhakypov, E. Kayacan, A. Sabanovic, "Assistive Manual Non-Contact Machining for Industrial and Surgical Operations", *Submitted to the 13th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2014
- E. A. Baran, T. E. Kurt, A. Sabanovic, "Lightweight Design and Encoderless Control of a Miniature Direct Drive Linear Delta Robot",

Proceedings of the 8th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, November 2013

- E. A. Baran, O. Ayit, V. B. Santiago, S. L. D. Guerra, A. Sabanovic, "A Self Optimizing Autofocusing Scheme for Microscope Integrated Visual Inspection Systems", *Proceedings of the 39th International Conference on Industrial Electronics (IECON)*, Vienna, Austria, November 2013
- E. A. Baran, E. Golubovic, T. E. Kurt, A. Sabanovic, "Constant Velocity Control of a Miniature Pantograph with Image Based Trajectory Generation" *Proceedings of the Asian Control Conference (ASCC)*, Istanbul, TURKEY, June 2013 (Finalist for "Best Application Paper Award" among over 700 submissions)

The following papers include results used in the thesis published/submitted in collaboration with other authors:

- E. Golubovic, E. A. Baran, A. Sabanovic, "Contouring Controller for Precise Motion Control Systems", *Automatika*, Vol. 54, No. 1, pp. 19-27, 2013 (SCI)
- A. Sabanovic, K. Ohnishi, D. Yashiro N. Sabanovic and E. A. Baran, "Motion Control Systems with Network Delay", *Automatika*, Vol. 51, No. 2, pp. 119-126, 2010 (SCI)
- A. Kuzu, E. A. Baran, S. Bogosyan, M. Gokasan, A. Sabanovic, "Performance Comparison of Compression Techniques Used in Bilateral Control", *Proceedings of the 39th International Conference on Industrial Electronics (IECON)*, Vienna, Austria, November 2013
- C. O. Saglam, E. A. Baran, A. Sabanovic, "Model Following Control with Discrete Time SMC for Time-Delayed Bilateral Control Systems" *Proceedings of the International Conference on Mechatronics (ICM)*, Istanbul, TURKEY, April 2011

Other papers of the author not referred in the content covered by the thesis:

- A. Kuzu, E. A. Baran, S. Bogosyan, M. Gokasan, A. Sabanovic, "Wavelet Packet Transform Based Compression for Bilateral Teleoperation", *Submitted to Journal of Systems and Control Engineering* (SCI)
- A. T. Naskali, E. D. Kunt, E. A. Baran, A. O. Nergiz, A. Sabanovic, "Software Framework for High Precision Motion Control Applications", *Submitted to Automatika* (SCI)

- E. A. Baran, E. Golubovic, A. Sabanovic, "Functional Observers for Motion Control Systems" *Automatika*, Vol. 54, No. 2, pp. 231-241, 2013 (SCI)
- K. Sendur and E. Baran, "Near-Field Power Transmission of Dipole Nano-Antennas" *Applied Physics B*, vol. 96, pp. 325-335, 2009 (SCI)
- T. Uzunovic, E. A. Baran, E. Golubovic, A. Sabanovic, "Three Dimensional Contour Tracking Control of a Parallel Manipulator: Comparison of Two Control Techniques", *Accepted for Publication in International Symposium on Industrial Electronics (ISIE)*, Istanbul, Turkey, June 2014
- Z. Zhakypov, T. Uzunovic, A. O. Nergiz, E. A. Baran, E. Golubovic, A. Sabanovic, "Desktop Microfactory for High Precision Assembly and Machining", *Accepted for Publication in International Symposium on Industrial Electronics (ISIE)*, Istanbul, Turkey, June 2014
- T. Uzunovic, E. Golubovic, E. A. Baran, A. Sabanovic, "Configuration Space Control of the Parallel Delta Robot with the Neural Network Based Inverse Kinematics", *Proceedings of the 8th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, Turkey, November 2013
- A. Kuzu, E. A. Baran, S. Bogosyan, M. Gokasan, A. Sabanovic, "WPT Based Compression for Bilateral Control", *Proceedings of the 39th International Conference on Industrial Electronics (IECON)*, Vienna, Austria, November 2013
- E. A. Baran, A. Sabanovic, "Predictive Input Delay Compensation for Motion Control Systems" *Proceedings of the International Conference on Advanced Motion Control (AMC)*, Sarajevo, BIH, March 2012
- E. Golubovic, I. S. M. Khalil, A. O. Nergiz, E. A. Baran, A. Sabanovic, "Design and Control of Laser Micromachining Workstation" *Proceedings of the International Conference on Advanced Motion Control (AMC)*, Sarajevo, BIH, March 2012
- E. A. Baran, A. Sabanovic, "A New Functional Observer to Estimate Velocity, Acceleration and Disturbance for Motion Control Systems" *Proceedings of International Symposium on Industrial and Electronics (ISIE)*, Bari, ITALY, July 2010

- H. Bilen, M. A. Hocaoglu, E. A. Baran , M. Unel, D. Gozuacik, "Novel Parameter Estimation Schemes in Microsystems" Proceedings of International Conference on Robotics and Automation (ICRA), Kobe, JAPAN, May 2009
- K. Sendur, O. Karabasoglu, E. Baran, and G. Kiziltas, "Optimization of Plasmonic Nano-Antennas" Proceedings of Material Research Society Symposium (MRS), Pittsburgh, USA, March 2008