Dynamics and Control of Free-Flying Manipulators Capturing Space Objects

Seyed Ali Akbar Moosavian

Department of Mechanical Engineering
McGill University
Montreal, Canada

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements of the degree of
Doctor of Philosophy

June, 1996

ISBN 0-612-19753-0

To

the memory of my father,

my beloved mother,

my dear wife, and sons.

Abstract

Based on the *barycentric vector* and *direct path* approaches, the kinematics of a multiple arm space robotic system is developed, and the differences between the two formulations are discussed. Applying the general Lagrangian formulation, a concise explicit model of the system dynamics is derived, and the specific characteristics of space robotic systems as compared to fixed-base manipulators are discussed.

Coordination between a spacecraft and its multiple end-effectors, based on planned trajectories, is investigated in capturing a moving space object. Two *model-based control algorithms*, based on an Euler angle and an Euler parameter description of the orientation, are proposed as well as a *transpose Jacobian controller*. Simulation results are presented to evaluate the developed controllers and the planning strategy, in both planar and three-dimensional maneuvers.

To control coordinated motions of space robotic systems, a new *Modified Transpose Jacobian* (MTJ) controller is presented which yields an improved performance over the standard algorithm. Simulation results show that the performance of the MTJ law is comparable to that of model-based algorithms, even though it requires a reduced computational effort.

To manipulate a captured object by multiple manipulators, a new *Multiple Impedance Control* (MIC) algorithm is developed which enforces an identical controlled impedance on each participating manipulator, on the manipulated object, and (in space) on the free-flying spacecraft. The similarities and differences between the developed MIC law and other force/impedance controllers are investigated, and simulation results are presented.

Résumé

En utilisant les approches de vecteurs baricentres et de trajectoires directes, la cinématique d'un système robotique spatial à plusieurs bras est développé, et une discussion sur la différence entre ces deux formulations est présentée. De plus, en recourant à la formulation générale de Lagrange, un modèle dynamique explicite et concis du système est présenté, de même que ses caractéristiques spécifiques qui sont comparées à celles de robots à base fixe.

La coordination entre le satellite et ses organes terminaux, pour des trajectoires planifiées, est étudiée lors de la capture d'un objet en mouvement. Deux systèmes d'asservissement utilisant un modèle de référence ont été développés, l'un utilisant une description de l'orientation par les angles d'Euler et l'autre par les paramètres d'Euler. Un système de commande du type de la matrice Jacobienne transposée fut aussi développé. Des résultats de simulation sont présentés afin d'évaluer ces trois systèmes d'asservissement lors de maneuvres planaires et tri-dimensionnelles.

Afin de permettre des mouvements coordonnnés d'un système robotique spatial, un nouveau modèle d'asservissement utilisant une matrice *Jacobienne Transposée Modifiée* (JTM) est présenté et résulte en des gains de performance par rapport à un algorithme standard. Des résultats de simulation démontrent que les performances de la loi JTM sont comparables à celles des algorithmes utilisant des modèles de référence , même si le nombre de calculs est réduit.

Pour manipuler un objet à l'aide de plusieurs manipulateurs, un nouvel algorithme de *Commande à Impédance Multiple* est développé. Celui-ci s'appuie sur une commande à impédance sur chaque manipulateur concerné, sur l'objet manipulé et (dans l'espace) sur le satellite servant de base. Une étude comparative entre cette loi et d'autres systèmes d'asservissement Force/Impédance est présentée, de même que des résultats de simulation.

In The Name of God

Acknowledgments

Completion of this thesis was conceivable with the generous help of my advisor, E. G. Papadopoulos. I am truly thankful for his understanding, encouragement, guidance, and brilliant comments throughout my research, as well as reviewing the manuscript. I learned a lot from our discussions during regular weekly meetings, and enjoyed experiencing his physical intuitions in solving theoretical problems. It was a great pleasure to work with him.

The support of this work by the Iran Ministry of Higher Education, and the Natural Sciences and Engineering Council of Canada (NSERC) is gratefully acknowledged.

I enjoyed attending several lectures in the Departments of Mechanical and Electrical Engineering, and my sincere gratitude goes to the whole faculty. Working at the Centre for Intelligent Machines was a wonderful experience, and I would like to thank the staff, secretaries, and all fellow students for providing such a pleasant environment. Special thanks are due to my colleagues Daniel Rey, Michael Glaum, Farzam Ranjbaran, and Eric Martin for their careful review of the manuscript, useful suggestions, and French translation of the abstract. Finally, I would like to express my appreciation to my wife Anoosheh for her great support and patience during this work. My sons are too small to read this, but I feel obliged to thank them for letting me keep working on this research.

Contents

Abbreviations viii

List of Symbols ix

List of Figures xvii

List of Tables xxi

1 Introduction 1

1.1 Motivation 1

1.2 Background 2

1.2.1 Manipulator Control 3

1.2.2 Space Robotics 6

1.3 Structure of the Work 9

1.3.1 Objectives 9

1.3.2 Research Tools 10

1.3.3 Thesis Outline 11

1.4 Contributions 13

2 Kinematics of Space Free-Flyers with Multiple Manipulators 15

2.1 Introduction 15

2.2 The Barycentric Vector Approach 16

2.2.1 Frame Assignment and Position Analysis 16

2.2.2 Definition of Barycentric Vectors 17

2.2.3 Velocity Analysis 19

2.2.4 Jacobian Matrix Associated with some Point and Link 20

2.3 The Direct Path Method 22

2.3.1 Frame Assignment and Position Analysis 22

2.3.2 Velocity Analysis 24
2.3.3 Jacobian Matrix Associated with some Point and Link....24
2.4 Discussion and Conclusions....25

3 Dynamics of Space Free-Flyers with Multiple Manipulators....27
3.1 Introduction....27
3.2 General Lagrangian Formulation....28
3.2.1 Kinetic Energy Calculations....28
3.2.1.1 Analysis Based on Barycentric Vector Kinematics....29
3.2.1.2 Analysis Based on Direct Path Kinematics....30
3.2.1.3 Comparison Between the Obtained Results....31
3.2.2 Equations of Motion via the Direct Path Approach....33
3.2.2.1 Preliminary Derivations....34
3.2.2.2 Mass Matrix....36
3.2.2.3 Vector of Nonlinear Terms....36
3.2.2.4 Vector of Generalized Forces....37
3.3 Supplementary Issues....38
3.3.1 Quasi-Coordinate Formulation....38
3.3.1.1 Problem Statement....38
3.3.1.2 Equations of Motion in terms of Quasi-Coordinates....39
3.3.2 Using Euler Parameters as Orientational Coordinates....41
3.3.2.1 Basic Definitions....41
3.3.2.2 Constrained Equations of Motion....43
3.3.2.3 Independent System of Equations....43
3.3.3 Dynamics Characteristics of SFFR....45
3.4 Generation of Symbolic Code for Dynamics....47
3.4.1 Symbolical vs. Numerical Code Generation....47
3.4.2 Description of the Code....50
3.4.3 Verification Procedure....51
3.5 Summary....53

4 Coordinated Motion Control of Multi-Arm Space Free-Flyers 55
4.1 Introduction.............55
4.2 Trajectory Planning.............56
4.2.1 Spacecraft Position and Orientation Trajectories.............57
4.2.2 Manipulator Motion Trajectories.............60
4.3 Control Algorithms Design.............60
4.3.1 Model-Based Control Design.............62
4.3.1.1 Using Euler Angles (MB1).............62
4.3.1.2 Using Euler parameters (MB2).............64
4.3.2 Transpose Jacobian Algorithm.............68
4.4 Simulation Results.............69
4.4.1 Code Implementation and Verification.............70
4.4.2 Example 1: Planar Motion.............70
4.4.2.1 Symmetric vs. Non-Symmetric Grasp.............74
4.4.2.2 Maximum Desired Acceleration and Deceleration of the Spacecraft.............74
4.4.2.3 Application of Alternative Control Algorithms.............76
4.4.3 Example 2: Three-Dimensional Maneuver.............78
4.4.3.1 System Description.............79
4.4.3.2 Comparison and Discussion.............82
4.5 Summary and Conclusions.............86
5 Modified Transpose Jacobian Control and its Application to Space Robotics 89
5.1 Introduction.............89
5.2 MTJ Control Law.............90
5.2.1 Motivation.............90
5.2.2 Derivation of MTJ Control Law.............92
5.3 Simulations and Comparisons.............95
5.3.1 Example 1: Two-Link Fixed-based Manipulator.............95
5.3.2 Example 2: Multiple Arm Space Free-Flying Robotic System.............101
5.4 Summary and Conclusions.............103

6 Multiple Impedance Control and its Application to Space Robotics 107

6.1 Introduction 107

6.2 Basic Concepts 108

6.2.1 Problem Statement and Task Definition 108

6.2.2 Application of Alternative Control Strategies 109

6.3 Multiple Impedance Control Law 112

6.3.1 General Formulation 113

6.3.2 Error Analysis 120

6.3.3 Contact Force Estimation 121

6.3.4 Case Study: A Comparative Analysis (Single Manipulator) 122

6.3.4.1 Dynamics Model 122

6.3.4.2 Controllability of the system 123

6.3.4.3 Root locus analysis 124

6.3.4.4 Simulation Results 126

6.3.4.5 Conclusions and Discussion of the Comparative analysis 129

6.3.5 Example: Cooperative Object Manipulation 130

6.3.5.1 Task Definition and Dynamics Modelling 130

6.3.5.2 Simulation Results and Discussions 132

6.4 Application of the MIC to Space Robotics 136

6.4.1 Basic Formulation 136

6.4.2 Error Analysis 142

6.5 Concluding Remarks 143

7 Conclusions and Suggestions 145

7.1 Conclusions 145

7.2 Suggestions for Future Research 148

Bibliography 151

Appendix A Three Formats Used in Dynamics Modelling 157

Appendix B Case Study: Transfer Functions 163

Appendix C Case Study: Root Locus Analysis 169

Appendix D Case Study: Simulation Results 175

Abbreviations

CM	Center of mass.
D-H	Denavit-Hartenberg notation.
des	Desired value (appears as subscript).
diag(a_1, ..., a_n)	Diagonal elements of a diagonal matrix.
DOF	Degrees of freedom.
EVA	Extra Vehicular Activities.
LHS	Left hand side.
MB	Model-based, a control algorithm.
MB1	Euler angle model based algorithm.
MB2	Euler parameter model based algorithm.
MIC	Multiple Impedance Control.
MTJ	Modified transpose Jacobian, a control algorithm.
MTJ1	The MTJ without using the error second derivative.
MTJ2	The MTJ when using the error second derivative.
OIC	Object Impedance Control.
PD	Proportional-Derivative (controller).
PID	Proportional-Integral-Derivative (controller).
RHS	Right hand side.
RCC	Remote Centre Compliance.
SFFR	Space Free-Flying Robot.
TJ	Transpose Jacobian, a control algorithm.

List of Symbols

$\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, $\mathbf{w}$	Elements of state-space representation of a single input linear system.
$\mathbf{a}$	A vector which describes a single constraint; $\mathbf{a}^T\,\dot{\mathbf{q}} = 0$.
$\mathbf{a}_1$, $\mathbf{a}_2$	Maximum acceleration, and deceleration (respectively) for planning spacecraft trajectories.
a_1, a_2, a_3	Typical terms of the system kinetic energy expression.
a_i, b_i, c_i	Coefficients of a second order algebraic equation.
$\mathbf{b}_c$	Damping matrix of an RCC unit.
b_i	Damping coefficient between the i-th mass and the next one, in a unilateral study.
$\mathbf{C}$	A vector which contains all gravity and nonlinear velocity terms of the dynamics model, where $\hat{\mathbf{C}}$ or $\tilde{\mathbf{C}}$ corresponds to the one in the task space, and a superscript "(i)" refers to the i-th manipulator.
$\breve{\mathbf{C}}$	Another representation of C; $\mathbf{C} = \breve{\mathbf{C}}\,\dot{\mathbf{q}}$.
$\hat{\mathbf{C}}$	The controllability matrix of a linear system.
$\mathbf{C}_1$, $\mathbf{C}_2$	A matrix, and a vector part of C; $\mathbf{C} = \mathbf{C}_1\,\dot{\mathbf{q}} + \mathbf{C}_2$.
C_i	Center of mass of the i-th body.
$\mathbf{E}$	A 3×4 matrix which relates angular velocity to the rate of Euler parameters; $\mathbf{E} = 2\begin{bmatrix}\breve{\mathbf{E}} & -\boldsymbol{\varepsilon}\end{bmatrix}$.
$\breve{\mathbf{E}}$	A 3×3 matrix defined as a block in E, $\breve{\mathbf{E}} = \eta\mathbf{1} + [\boldsymbol{\varepsilon}]^{\times}$.
$\mathbf{e}$, $\tilde{\mathbf{e}}$	Vector of tracking errors, where a subscript is used for a particular variable, e.g. $\mathbf{e}_{\omega}$ is the error in angular velocity, and a superscript "i" corresponds to the i-th manipulator.
$\mathbf{e}_i^{(m)}$, $\mathbf{l}_i^{(m)}$, $\mathbf{r}_i^{(m)}$	Body-fixed vectors which describe position of CM for the augmented body i, joint i, and joint i+1 with respect to C_i, respectively.
e_{max}, $\dot{e}_{max}$	Sensitivity thresholds, in the MTJ law.

$\tilde{\mathbf{e}}_k^{(m)}$, $\tilde{\mathbf{l}}_k^{(m)}$, $\tilde{\mathbf{r}}_k^{(m)}$	Barycentric vectors which describe position of C_i, joint i, and joint i+1 with respect to the CM of the augmented body i, respectively.
$\mathbf{F}_{0,p}$	The p-th external force/moment applied on the spacecraft.
$\mathbf{F}_{add}$	An additional force to create a couple (torque) by two end-effectors holding an object, for controlling its orientation in a pivoted grasp condition.
$\mathbf{F}_c$	A 6×1 vector which contains the forces/moments applied on an acquired object due to contact with the environment.
$\hat{\mathbf{F}}_c$	Estimated value of contact force $\mathbf{F}_c$.
$\mathbf{F}_e$	A 6n×1 vector which contains all end-effector forces/torques applied on an acquired object, where $\mathbf{F}_e^{(i)}$ is a 6×1 vector corresponding to the i-th end-effector.
$\mathbf{F}_{e_{req}}$	Required end-effector forces/torques to be applied on an acquired object, where $\mathbf{F}_{e_{req}}^{(i)}$ is a 6×1 vector corresponding to the i-th end-effector.
$\mathbf{F}_G$	Required force for moving an internal angular momentum source along with the acquired object motion.
$\mathbf{F}_{int}$	A 6×1 vector of internal forces/moments, to be generated in an acquired object.
$\mathbf{F}_{i,p}^{(m)}$	The p-th external force/moment applied on the i-th body of the m-th manipulator.
$\mathbf{F}_o$	A 6×1 vector which contains external forces/moments (other than contact and end-effector ones) applied on an acquired object.
$\mathbf{F}_\omega$	A 6×1 vector which contains nonlinear velocity terms in an acquired object dynamics equations.
$F_{i_{filtered}}$	A low-pass filtered value of F_i.
$\mathbf{f}_c$, $\mathbf{n}_c$	Resultant force (torque) applied on an acquired object due to contact, where $\mathbf{n}_c$ includes moment of $\mathbf{f}_o$ about the object CM.
$\mathbf{f}_e^{(i)}$, $\mathbf{n}_e^{(i)}$	The i-th end-effector force (torque) exerted on an acquired object.
$\mathbf{f}_o$, $\mathbf{n}_o$	Vector of external forces (torques), other than contact and end-effector ones, applied on an acquired object (including gravity effects), where $\mathbf{n}_o$ includes moment of $\mathbf{f}_o$ about the object CM.
${}^0\mathbf{f}_s$, ${}^0\mathbf{n}_s$	The net external force and torque applied on the spacecraft, expressed in its own body-fixed frame.
$\mathbf{G}$	A 6×6n grasp matrix which maps the vector of all end-effector forces/torques to an acquired object dynamics equations.
$\mathbf{G}^{\#}$	A weighted pseudoinverse of the grasp matrix $\mathbf{G}$.

$\mathbf{g}$	A function which describes the grasp constraint, where a superscript "i" refers to that of the i-th manipulator.
$G(s)$	A transfer function which relates the output to the input of a linear system, a subscript is used to specify the applied control algorithm.
$\mathbf{H}$	The positive definite mass matrix of the system, where $\hat{\mathbf{H}}$ or $\tilde{\mathbf{H}}$ corresponds to the one in the task space, and a superscript "i" refers to the i-th manipulator.
$\mathbf{h}(t)$	A modifying term added to the TJ control algorithm.
i_f	Number of applied force/torque vectors on a body.
i_s	Number of system constraints.
$\mathbf{J}_C$	A square Jacobian matrix which relates the output speeds to the generalized ones, where a superscript "i" corresponds to the i-th manipulator..
$\mathbf{J}_{i,p}^{(m)}$	A Jacobian matrix which relates the linear velocity of an arbitrary point P (on the i-th body of the m-th manipulator) and the angular velocity of the corresponding body, to the generalized speeds, where $\mathbf{J}_{0,p}$ is for P on the spacecraft, and $\mathbf{J}_0$ when P is the spacecraft CM.
$\mathbf{J}_Q$	An N×N Jacobian matrix which relates the vector of actuator forces/toques to the vector of generalized forces.
$\mathbf{J}_L^{(i)}$, $\mathbf{J}_A^{(i)}$	Linear and angular parts of Jacobian matrix for the i-th link of a fixed-base manipulator.
$\mathbf{K}_p$, $\mathbf{K}_d$	Control gain matrices.
$\mathbf{k}$	A 3×1 unit vector along spacecraft axis of rotation.
$\mathbf{k}_c$	Stiffness matrix of an RCC unit.
$\tilde{\mathbf{k}}_p$, $\tilde{\mathbf{k}}_d$, $\tilde{\mathbf{M}}_{des}$	Block diagonal N×N control gain and desired mass matrices, composed of the corresponding 6×6 matrices which define the impedance law for the acquired object.
k	Regulating factor, in the MTJ algorithm.
k_i	Stiffness coefficient between the i-th mass and the next one, in a unilateral study.
k_w	Stiffness coefficient of an obstacle located at x_w, in a unilateral study.
K	Number of all joints, for a multiple arm robotic systems, $\sum_{m=1}^{n} N_m$.
$\mathbf{L}_G$	Angular momentum of an internal source about the acquired object CM.
$\mathbf{L}_s$	Angular momentum of an internal source about its own CM.
$l_{i_{free}}$	Free length of a spring, k_i.

$\mathbf{l}_{RCC}$	A vector defining an RCC's free-length in different directions.
$\mathbf{M}$	A 6×6 mass matrix for an acquired object.
$\mathbf{M}_{des}$	An acquired object desired mass matrix, in the impedance law.
$\mathbf{M}_G$	Required moment for moving an internal angular momentum source along with the acquired object motion.
M	Total mass of a space robotic system.
m_0, $\mathbf{I}_0$	Mass and inertia dyad of the spacecraft with respect to its CM.
$m_i^{(m)}$, $\mathbf{I}_i^{(m)}$	Mass and inertia dyad of the i-th body of the m-th manipulator with respect to its CM.
m_i, ${}^0\mathbf{I}_i^{CM_i}$	Mass of the i-th link of a single fixed-base manipulator, and its inertia matrix with respect to its CM expressed in the fixed frame, where $\hat{m}_i$ corresponds to a given or estimated value.
m_{obj}, $\mathbf{I}_G$	An acquired object mass, and its moment of inertia about CM.
m_s	Mass of an internal angular momentum source which is not included in the acquired object mass m_{obj}.
n	Number of manipulators or appendages, for a system of multiple manipulators.
N	The system total degrees of freedom (DOF).
N_m	Number of joints (single DOF), for the m-th manipulator.
$\mathbf{P}$	A Jacobian matrix between the vector of generalized speeds and the time derivative of generalized coordinates vector; $\mathbf{P} = \Psi^T$.
$\mathbf{p}_s$	Linear momentum of an internal angular momentum source, inside an acquired object.
${}^j\mathbf{P}^j_{CM_i}$	The i-th link CM position vector with respect to the origin of j-th frame expressed in that frame.
$\mathbf{Q}$	Vector of generalized forces, where $\hat{\mathbf{Q}}$ or $\tilde{\mathbf{Q}}$ corresponds to the one in the task space, and a superscript "i" refers to the i-th manipulator.
$\mathbf{Q}_\delta^{(0)}$	A 3×1 vector which contains the generalized forces corresponding to the spacecraft orientation.
$\tilde{\mathbf{Q}}_{app}$	Applied controlling force (expressed in the task space), where a superscript "i" refers to the i-th manipulator; $\tilde{\mathbf{Q}}_{app} = \tilde{\mathbf{Q}}_m + \tilde{\mathbf{Q}}_f$.
$\tilde{\mathbf{Q}}_f$	Required force to be applied on the manipulated object by the end-effector, where a superscript "i" refers to the i-th manipulator.
$\tilde{\mathbf{Q}}_m$	Applied controlling force concerning the motion of the end-effector, where a superscript "i" refers to the i-th manipulator.
$\tilde{\mathbf{Q}}_{react}$	Reaction force (expressed in the task space) on the end-effector, where a superscript "i" refers to the i-th manipulator.

$\mathbf{Q}_r$	Vector of constraint forces/torques in dynamics equations.
$\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$	An N×1 vector of generalized coordinates, and its rate, where a superscript "i" corresponds to the i-th manipulator.
$\hat{\mathbf{q}}$, $\dot{\hat{\mathbf{q}}}$, $\ddot{\hat{\mathbf{q}}}$	A vector of controlled variables, and its rate, where $\hat{\mathbf{q}}_{des}$, $\dot{\hat{\mathbf{q}}}_{des}$, and $\ddot{\hat{\mathbf{q}}}_{des}$ refer to the desired ones, and a superscript "i" corresponds to the i-th manipulator.
$\hat{\mathbf{q}}_*$, $\dot{\hat{\mathbf{q}}}_*$	A linear combination of $\hat{\mathbf{q}}$ (or $\dot{\hat{\mathbf{q}}}$) values at different time steps.
$\mathbf{R}_{C_0}$, $\dot{\mathbf{R}}_{C_0}$, $\ddot{\mathbf{R}}_{C_0}$	Inertial position, velocity, and acceleration of the spacecraft CM, where the components are expressed as x_0, y_0, z_0, etc.
$\mathbf{R}_{CM}$, $\dot{\mathbf{R}}_{CM}$, $\ddot{\mathbf{R}}_{CM}$	Inertial position, velocity, and acceleration of the system CM, where the components are expressed as x_{CM}, y_{CM}, z_{CM}, etc.
$\mathbf{R}_p$, $\dot{\mathbf{R}}_p$	Inertial position and velocity of an arbitrary point P.
${}^0\mathbf{r}$	Relative position between the spacecraft CM and a point of interest on an acquired object at final time.
$\mathbf{r}_{C_i}$, $\mathbf{r}_{C_i}^{(m)}$	The CM position vector of the i-th body with respect to the spacecraft CM.
$\mathbf{r}_e^{(i)}$	Position vector of the i-th end-effector with respect to an acquired object CM.
$\mathbf{r}_p$	Position vector of point P with respect to the spacecraft CM
$\mathbf{r}_{p/C_i}$	Position vector of point P with respect to C_i.
$\mathbf{r}_s$	The CM position vector of an internal angular momentum source with respect to an acquired object CM.
$\mathbf{S}_0$	A 3×3 matrix, which relates the spacecraft angular velocity to the Euler angle rates, where $\mathbf{S}_{obj}$ is the one for an acquired object.
s_i	A root of the characteristic equation.
$\mathbf{T}_0$, $\mathbf{T}_j^{(k)}$	Rotation matrices between body-fixed frames and the inertial frame.
${}^j\mathbf{T}_i$	Rotation matrix between the i-th frame and the j-th one.
$\mathbf{T}_{des}$	Rotation matrix which corresponds to the desired orientation.
$\mathbf{T}_e$	Rotation matrix which relates the error between the desired and current attitude.
T	Kinetic energy of the whole system.
T_0, T_1, T_2	Different terms defined in the system kinetic energy expression.
t_1	The time at which acceleration segment ends, in planning the spacecraft desired trajectory.
t_2	The time at which the deceleration segment starts, in planning the spacecraft desired trajectory.
t_f	A motion final time, for trajectory planning.

t_n	Current time at the n-th time step, where Δt_n denotes the step size.
t_r	The time at which an acquired object enters the reachable (fixed-base) workspace of an end-effector.
t_*	The instant at which $\dot{V}(\hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}})$ (the time derivative of a Lyapunov function) vanishes.
$\mathbf{U}_{f_c}$	An N×6 matrix, composed of $(n+1)$ 6×6 identity matrices.
$\mathbf{u}$	An auxiliary control signal, where a subscript is used for a particular variable, e.g. $\mathbf{u}_{\mathbf{R}_0}$ denotes the one which corresponds to the spacecraft CM.
$u(t)$	A single input to the system.
${}^0\mathbf{v}_0(0)$	Initial velocity of the spacecraft CM expressed in the body-fixed frame at initial time.
$\tilde{\mathbf{v}}_{ki}^{(m)}$	A general barycentric vector.
$\mathbf{v}_s$	Inertial linear velocity of an internal angular momentum source CM.
$V(\hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}})$, $\dot{V}(\hat{\mathbf{q}}, \dot{\hat{\mathbf{q}}})$	A candidate for Lyapunov function, and its rate, where a subscript refers to its value at specific time step, e.g. V_n for the n-th time step.
XYZ	The inertial frame of reference.
$\mathbf{xyz}_i$	The i-th link body-fixed frame.
$\mathbf{x}$	A 6×1 vector which contains the CM position, and Euler angles of an acquired object, where $\dot{\mathbf{x}}$ and $\ddot{\mathbf{x}}$ are its rates, and $\mathbf{x}_{des}$, etc. are the desired ones.
$\tilde{\mathbf{x}}$, $\dot{\tilde{\mathbf{x}}}$, $\ddot{\tilde{\mathbf{x}}}$	A vector of controlled variables, and its rates, where $\tilde{\mathbf{x}}_{des}$, $\dot{\tilde{\mathbf{x}}}_{des}$, and $\ddot{\tilde{\mathbf{x}}}_{des}$ refer to the desired ones, and a superscript "i" corresponds to the i-th manipulator.
${}^0\mathbf{x}_0(t)$	Desired trajectory for the spacecraft CM position expressed in the body-fixed frame at initial time.
$\mathbf{x}_0(t)$	Desired trajectory for the spacecraft CM position, in the inertial frame.
$\mathbf{x}_0(0)$	Inertial position of the spacecraft CM at initial time.
$\mathbf{x}_E^{(m)}$, $\dot{\mathbf{x}}_E^{(m)}$	The m-th end-effector inertial position, and velocity vector; $\mathbf{x}_E^{(m)}$ = $(x_E^{(m)}, y_E^{(m)}, z_E^{(m)})$, etc.
${}^0\mathbf{x}_f$, ${}^0\mathbf{v}_f$	Desired final position, and velocity for the spacecraft CM, expressed in the body-fixed frame at initial time.
$\mathbf{x}_G, \dot{\mathbf{x}}_G, \ddot{\mathbf{x}}_G$	Inertial position, velocity, and acceleration of an acquired object CM.
${}^0\mathbf{x}_{obj}^0(0)$, ${}^0\mathbf{v}_{obj}^0(0)$	Position and velocity of an acquired object as measured with respect to the spacecraft CM at initial time, expressed in the spacecraft frame.
$\mathbf{x}_i, \dot{\mathbf{x}}_i, \ddot{\mathbf{x}}_i$	Position, velocity, and acceleration of the i-th mass, in a unilateral study.

$\mathbf{y}$	An output vector in state-space representation of a linear system.
$\mathbf{z}_i^{(m)}$	A unit vector along axis of rotation of the i-th joint of the m-th manipulator (for single DOF joints); ${}^j\mathbf{z}_j^{(k)} \equiv (0,0,1)^T$.
${}^0\mathbf{z}_i$	A unit vector along i-th joint axis expressed in the base frame.

Greek Symbols:

$\alpha_i^{(m)}$	An angle between the z-axis of the i-th frame and the one of the previous frame, according to D-H convention.
$\boldsymbol{\delta}_0$	A set of Euler angles which describes the spacecraft attitude; $\boldsymbol{\delta}_0 = (\alpha_0, \beta_0, \gamma_0)$.
$\boldsymbol{\delta}_E^{(m)}$	A set of Euler angles which describes the m-th end-effector orientation; $\boldsymbol{\delta}_E^{(m)} = (\alpha_E^{(m)}, \beta_E^{(m)}, \gamma_E^{(m)})$, and becomes a single angle $\delta_E^{(m)}$ in planar motion.
$\boldsymbol{\delta}_{obj}$	A set of Euler angles which describes an acquired object attitude.
Δt	A time step.
δ	Logarithmic decrement.
δ_{ji}	The Kronecker delta.
$\boldsymbol{\varepsilon}$	Three first components of the Euler parameters; $\boldsymbol{\varepsilon} = \mathbf{k}\sin(\theta_0/2)$.
ε, $\dot{\varepsilon}$	Sensitivity thresholds, in the MTJ algorithm.
ζ	Damping ratio in a second-order differential equation.
η	The fourth component of the Euler parameters; $\eta = \cos(\theta_0/2)$.
$\boldsymbol{\theta}^{(m)}$	An $N_m \times 1$ column vector which contains the joint angles of the m-th manipulator, where $\theta_i^{(m)}$ refers to its i-th component (joint).
$\boldsymbol{\theta}$	A $K \times 1$ column vector which contains all joint angle vectors, $\left(\boldsymbol{\theta}^{(1)T}, \boldsymbol{\theta}^{(2)T}, \cdots, \boldsymbol{\theta}^{(n)T}\right)^T$.
θ_0	A rotation angle about the spacecraft axis of rotation which is used for the spacecraft Euler parameter determinations, also determines the spacecraft attitude in planar motion.
$\boldsymbol{\kappa}$	A set of Euler parameters which describes the spacecraft orientation; $\boldsymbol{\kappa} = \left(\boldsymbol{\varepsilon}^T, \eta\right)^T$.
Λ	A Lagrange multiplier for a single constraint, where Λ_k is for the k-th one.
λ_i	The i-th eigenvalue of a matrix.

$\mu_i^{(m)}$	Ratio of outboard mass (after the i-th joint of the m-th manipulator) with respect to the total mass.
$\boldsymbol{\nu}$	Vector of generalized speeds.
$\boldsymbol{\rho}_{C_0}$	Position vector of the spacecraft CM, with respect to the system CM.
$\boldsymbol{\rho}_{C_i}$, $\boldsymbol{\rho}_{C_i}^{(m)}$	Position vector of C_i, with respect to the system CM.
$\boldsymbol{\rho}_P$	Position vector of point P, with respect to the system CM.
$\boldsymbol{\sigma}_i$, $\boldsymbol{\sigma}_i'$	Angular velocity derivatives with respect to generalized coordinates, and their rates.
$\boldsymbol{\tau}$	Vector of joint forces/torques.
$\boldsymbol{\Phi}$	A matrix which relates the time derivative of generalized coordinates vector to the vector of generalized speeds; $\dot{\mathbf{q}} = \boldsymbol{\Phi}\boldsymbol{\nu}$.
$\boldsymbol{\Psi}$	A matrix which relates the vector of generalized speeds to the time derivative of generalized coordinates vector; $\boldsymbol{\nu} = \boldsymbol{\Psi}\dot{\mathbf{q}}$
$\boldsymbol{\omega}_0$	Spacecraft angular velocity, and ${}^0\boldsymbol{\omega}_0 \equiv ({}^0\omega_{0_1}, {}^0\omega_{0_2}, {}^0\omega_{0_3})^T$ when expressed in its own body-fixed frame.
$\boldsymbol{\omega}_k^{(m)}$	Angular velocity of the k-th body of the m-th manipulator.
${}^m\boldsymbol{\omega}_E^{(m)T}$	Angular velocity of the m-th end-effector expressed in its own body-fixed frame.
$\boldsymbol{\omega}_{obj}$, $\dot{\boldsymbol{\omega}}_{obj}$	An acquired object angular velocity, and acceleration.
ω	Frequency of a trajectory.
ω_0	Frequency of a low-pass filter.

List of Figures

Figure 1.1: Concept of SFFRs, (a) The Orbital Servicing Vehicle, (b) The Extra Vehicular Astronaut Retriever. 2

Figure 2.1: A free-flying space robotic system with n manipulators....17

Figure 2.2: A free-flying space robotic system with n manipulators....23

Figure 4.1: Typical profiles of the planned parabolic trajectory....58

Figure 4.2: A planar three manipulator and appendage free-flyer, Ex. 1....71

Figure 4.3: (a) Joint angle histories for the two manipulators and the antenna, (b) Animated view of the maneuver....73

Figure 4.4: Applied torque on the spacecraft, (a) Symmetric grasp, (b) Non-symmetric grasp....74

Figure 4.5: The effect of acceleration/deceleration ratio, (a) Spacecraft thruster forces, (b) Applied torque on the spacecraft, (c) First end-effector positioning error....75

Figure 4.6: TJ compared to MB Control. (a) Tracking position errors for the first end-effector, (b) Thruster forces on the spacecraft, (c) Joint torques for the first arm....77

Figure 4.7: A three manipulator and appendage free-flyer, Ex. 2....78

Figure 4.8: Errors in spacecraft orientation encountering a non-physical singularity at time = 4.75 sec....81

Figure 4.9: The desired path for the spacecraft center of mass and the two end-effectors. 82
Figure 4.10: Tracking position errors for the first end-effector, (a) Model-Based Control, MB1. (b) Model-Based Control, MB2. (c) Transpose Jacobian Control. 83
Figure 4.11: Thruster forces on the spacecraft, (a) MB1, (b) MB2, (c) TJ, algorithm. 85
Figure 4.12: Applied torques on the spacecraft (left) and joint torques for the first end-effector (right), (a) MB1, (b) MB2, (c) TJ Algorithm....86
Figure 5.1: (a) A two-link planar manipulator, (b) Desired tracking path....96
Figure 5.2: Tracking errors for low-speed task, (a) TJ algorithm, (b) MTJ algorithm. 97
Figure 5.3: Tracking errors for high-speed task, (a) TJ algorithm, (b) MTJ algorithm. 97
Figure 5.4: Joint torques and tracking errors, (a) TJ with high gains, (b) MTJ, (c) MB, case 1, (d) MB, case 2....99
Figure 5.5: Joint torques in the presence of noisy feedback, (a) TJ, high gains, (b) MTJ, (c) MB, case 1, (d) MB, case 2. 100
Figure 5.6: Tracking position errors for the first end-effector, (a) TJ, (b) MB, (c) MTJ1, (d) MTJ2....102
Figure 5.7: Thruster forces (left) and applied torques on the spacecraft (right), (a) TJ, (b) MB, (c) MTJ1, (d) MTJ2. 104
Figure 5.8: Joint torques for the first manipulator, (a) TJ, (b) MB, (c) MTJ1, (d) MTJ2. 105
Figure 6.1: A simple spring, to visualize notion of impedance behavior....109
Figure 6.2: A model of performing a manipulation task by a single robot....109
Figure 6.3: Two robotic arms performing a cooperative manipulation task....114

Figure 6.4: The block diagram of the MIC algorithm....................................119
Figure 6.5: Performance of the MIC, (a) Object tracking error, (b) The applied controlling force, (c) The contact force....................................128
Figure 6.6: Performance of the OIC, (a) Object tracking error, (b) The applied controlling force, (c) The contact force....................................128
Figure 6.7: Two robotic arms, performing a cooperative manipulation task in planar motion....................................130
Figure 6.8: Application of the MIC in cooperative manipulation, (a) Error in object CM position and object orientation, (b) Velocity errors, (c) First manipulator joint torques, (d) Second manipulator joint torques, (e) Real value of the contact force, (f) Difference between the real value of contact force and estimated one....................................134
Figure 6.9: An object with an internal angular momentum source, manipulated by a multiple arm SFFR....................................138
Figure B.1: Block diagram for the open-loop system....................................164
Figure B.2: Block diagram for the MIC implementation....................................167
Figure B.3: Block diagram for the OIC implementation....................................168
Figure C.1: Root locus for the MIC law, $b_2 = 100$, (a) In contact (b) No contact.......170
Figure C.2: Root locus for the OIC law, $b_2 = 100$, (a) In contact (b) No contact........170
Figure C.3: Root locus for the MIC law, $b_2 = 10$, (a) In contact (b) No contact.........171
Figure C.4: Root locus for the OIC law, $b_2 = 10$, (a) In contact (b) No contact.........171
Figure C.5: Root locus for the OIC law, $b_2 = 10$, and no contact, (a) Larger Kp, (b) Larger Kd, (c) Larger mdes, (d) Larger Kd and mdes....................172
Figure C.6: Root locus for the MIC law, $b_2 = 0.0$, (a) In contact (b) No contact........173
Figure C.7: Root locus for the OIC law, $b_2 = 0.0$, (a) In contact (b) No contact........173
Figure D.1: Performance of the MIC, kd =700, (a) Object tracking error, (b) The applied controlling force, (c) The contact force....................................176

Figure D.2: Performance of the OIC, kd =700, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.................................176

Figure D.3: Performance of the MIC with actuator saturation limit, |Factuator| < 80, (a) Object tracking error, (b) The applied controlling force, (c) The contact force. 177

Figure D.4: Performance of the OIC with actuator saturation limit, |Factuator| < 80, (a) Object tracking error, (b) The applied controlling force, (c) The contact force. 177

Figure D.5: Performance of the MIC, kp =1000, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.................................178

Figure D.6: Performance of the OIC, kp =1000, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.................................178

Figure D.7: Performance of the MIC, b2 =10, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.................................179

Figure D.8: Performance of the OIC, b2 =10, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.................................179

Figure D.9: Performance of the OIC, for b2 =10, mdes =500, kd=1000, (a) Object tracking error, (b) The applied controlling force, (c) The contact force. 180

List of Tables

Table 3.1: The result of verification procedure....52

Table 4.1: Comparison of the required computational operations....69

Table 4.2-a: Spacecraft parameters and actuator limits, Ex. 1....71

Table 4.2-b: Manipulator parameters and joint actuator limits, Ex. 1....71

Table 4.3-a: Spacecraft parameters and actuator limits, Ex. 2....79

Table 4.3-b: Manipulator parameters and the joint actuator limits, Ex. 2....79

Table 5.1: Selected gains for alternative algorithms, Ex. 1....98

Table 5.2: Selected gains for alternative algorithms, Ex. 2....101

Table 6.1: Parameters for the system depicted in Figure 6.7....133

Chapter 1

Introduction

1.1 Motivation

As space commercialization materializes, space structures and satellites will proliferate. Extending the life of such systems, and therefore reducing the associated costs, will require extensive inspection, assembly, capture, repair and maintenance capabilities in orbit. Astronaut Extra Vehicular Activities (EVA) can be valuable in meeting these requirements. However, the cost of human life support facilities, the limited time available for the maneuver, and the high risks involved due to various hazards, are some serious restrictions for EVA. Therefore, it is expected that robotic devices will play an important role in future missions.

To increase the mobility of such robotic systems, Space Free-Flying Robots (SFFRs) in which manipulators are mounted on a thruster-equipped spacecraft, have been proposed (Bronez et al (1986), Reuter et al. (1988)), see Figure 1.1. Unlike fixed-based robots, the base body of SFFR is allowed to respond freely to dynamic reaction forces due to the arms motion. Hence, in order to control such a system, it is essential to consider the dynamic coupling between the arms and the base. Also it should be noted that the joint control torques are limited due to actuator weight constraints in space.

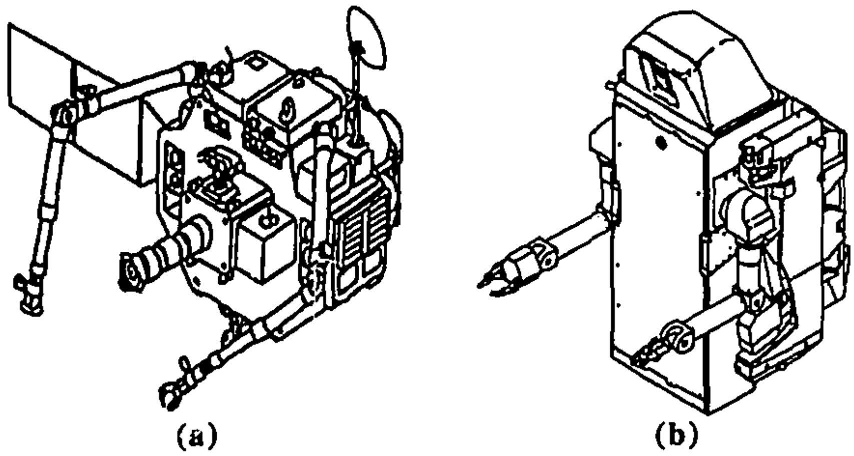


Figure 1.1: Concept of SFFRs, (a) The Orbital Servicing Vehicle, (b) The Extra Vehicular Astronaut Retriever.

Although dynamics modelling of SFFR is still an ongoing subject of research, control of these free-flying manipulators to perform precise tasks in space, has already received some attention. Control techniques for space manipulators can be classified in three different categories. In the first, both the position and attitude of the base are actively controlled (*free-flying mode*). In the second category neither of them is controlled (*free-floating mode*) and finally, in the third, only the base attitude is controlled. Clearly, a combination of these three modes can be employed during different phases of a mission. In this research work, the focus is on the free-flying mode, and more precisely on the coordination and control of the spacecraft and its multiple arms in capturing and manipulating space objects.

1.2 Background

Control of mechanical manipulators is a challenging task, because of the strong nonlinearities in the equations of motion. Different algorithms have been suggested to control the end-effector position, orientation, or force, since the early research in robotics. In this section, first a brief review of popular algorithms to control *fixed-base* manipulators is introduced. Next, a set of studies on the dynamics and control of *space free-flying robots* will be briefly reviewed.

1.2.1 Manipulator Control

Position Control. In this category, it is assumed that there is no force interaction between the end-effector and the environment while its position and orientation have to be controlled. Classic PID controllers at each joint of the manipulator are widely employed in industrial geared robots. Although these feedback controllers are designed on the basis of neglecting the dynamic coupling between the links, they can effectively control the system, (Arimoto and Miyazaki (1984), (Kawamura et al. (1988)). High gear ratios reduce the relative importance of the manipulator dynamics, but do not eliminate the requirement for an accurate system modelling, Leahy and Saridis (1989). The *Computed Torque Method* employs such a model to compensate for the nonlinearities, and result in a linearized error behavior[1]. Khosla and Kanade (1989), and An et al. (1989) presented two sets of experimental studies which compare the performance of the independent joint control schemes (e.g. classic PID) to the computed torque method, implemented on direct drive manipulators. These studies conclude the importance of compensating for the nonlinear *Coriolis* and *centrifugal* forces, even at low speeds of operation.

The application of *Model-Referenced Adaptive Control* to robotic manipulators is based on an adaptation algorithm which changes the controller gains so that the real output follows the referenced model output within an accuracy bound (Dubowsky and Des Forges (1979), Slotine and Li (1987)). Youcef-Toumi and Ito (1987) suggested *Time Delay Control* which is a model-referenced algorithm for systems with unknown dynamics. The basic function of the controller is to use observations of the system response to directly modify the control actions rather than adjusting the controller gains. Arimoto and Miyazaki (1984) proposed the *betterment process* which is based on a learning control approach and improves operation of a robot in the next cycle so that the motion trajectory converges

---

1- According to Craig (1989), the idea was first proposed by Paul (1972), and named as the Computed Torque Method by Markiewicz (1973) and Bejczy (1974).

eventually to the desired one. Therefore this algorithm can be applied when repetitive operations are to be performed.

The *Transpose Jacobian* (TJ) control is a computationally simple algorithm, which has been arrived at intuitively (Craig (1989)). The task error vector and its rate, both multiplied by relatively high gains, and by the Jacobian transpose matrix, result in commands that push the end-effector in a direction which tends to reduce the tracking error. In the case of using an approximate Jacobian, it has been shown that the damping matrix and the position gain matrix of this controller play an important role in determining the stability margin (Miyazaki et al. (1988)). The TJ algorithm does not fail when a singularity occurs (Chiaverini et al. (1990)), and can be applied to redundant manipulators (Asari et al. (1993)). An extended TJ control algorithm has been developed to improve the performance of mobile manipulator systems (Hootsmans and Dubowsky (1991)), and also to coordinate motion control of spacecraft/manipulator systems (Papadopoulos and Dubowsky (1991b)).

Force/Impedance Control. Position or motion control algorithms are not sufficient to control an end-effector's interaction with its environment. To control the interaction forces or the dynamic behavior of the manipulator during tasks involving contact, force and impedance control laws have been proposed. Raibert and Craig (1981) suggested the *Hybrid Position/Force Algorithm* to control end-effector position in some directions, and its contact forces in the remaining directions. Using wrist force sensors, and defining a compliance selection matrix to determine position or force priorities in orthogonal directions, a hybrid control architecture is implemented for tasks which require contact with the environment. Hayati (1986) extended this approach to a system of multiple manipulators. Khatib (1987) presented the *Operational Space Formulation* for motion and force control of robotic manipulators. Defining generalized task specification matrices for motion and contact forces, and employing a nonlinear dynamic decoupling approach, he presents a control architecture with slow computation of dynamics, and a fast servo level to compute the control command. Whitney (1987) compared different strategies in robot force control, and discussed some unsolved problems.

Nakamura et al. (1987) discussed the mechanics of coordinative manipulation by multiple robotic mechanisms, taking the dynamics of the object being moved into consideration. Assuming a frictional grasp, they propose a computational procedure to attain optimal internal forces. Tarn et al. (1987) presented a closed chain formulation for the dynamic control of two cooperative manipulators with equal degrees of freedom. Hayward and Hayati (1988) discussed various issues in the design of a multi-manipulator control system, and developed an environment for the programming and control of cooperative manipulators.

For a single manipulator in dynamic interaction with its environment, Hogan (1985) proposed the *Impedance Control* that regulates the relationship between end-effector position and force. Starting from basic concepts, a method is suggested for choosing an appropriate manipulator impedance. Goldenberg (1988) proposed an implementation of a combined impedance and force control, to exert a desired force on the environment, and at the same time, generate a desired relationship between this force and the relative location of the point of interaction (contact) with respect to the commanded manipulator location. Using an exact model of the manipulator, the algorithm is developed based on feedback and feedforward control methods. Seraji and Colbaugh (1993) presented two adaptive schemes to make impedance control capable of tracking a desired contact force, which has been described as the main shortcoming of impedance control in an unknown environment. The first scheme is based on an on-line reference position generating procedure, as a function of force tracking errors. The second one is developed based on an on-line parameter estimation procedure to obtain the environmental unknowns, and compute the proper reference position for tracking a desired contact force.

As an extension of Hogan's impedance control concept, Schneider and Cannon (1992) developed the *Object Impedance Control* (OIC) for multiple robotic arms manipulating a common object. A combination of feedforward and feedback control is employed to make the object behave like a reference impedance. Meer and Rock (1995) tried to extend OIC to a class of flexible objects. They realized that attempting to apply this controller when a flexible object interacts with its environment may lead to instability. Based on the analysis of a representative system, they suggest that in order to solve the instability problem, one should either increase the desired mass parameters or filter and lower the frequency content of the estimated contact force.

1.2.2 Space Robotics

Dynamics and control of SFFRs, unlike those for long reach space manipulators, are usually investigated under the assumption of rigid elements. This assumption characterizes the following research studies on SFFRs.

Kinematics and Dynamics. Vafa and Dubowsky (1987) described the kinematics and dynamics of a free-floating space manipulator system, using the *Virtual Manipulator Approach*. No external forces act on the system, and so the system center of mass is fixed in inertial space, enabling them to represent a free-floating system by one with a virtual fixed base. Papadopoulos and Dubowsky (1991a) employed a *barycentric vector approach*, to study kinematics and dynamics of a single arm SFFR in free-floating mode. Taking the center of mass of the whole system as a representative point for the translational motion, and using barycentric vectors which reflect both geometric configuration and mass distribution of the system, results in a decoupling of the total linear and angular motion from the rest of the equations. Umetani and Yoshida (1987) presented a *Generalized Jacobian Matrix* for a free-floating system. Assuming that no external forces are applied on a rigid robotic system with revolute joints, they derive a generalized Jacobian matrix which reflects both momentum conservation laws and kinematic relations. The proposed generalized Jacobian matrix converges to the conventional Jacobian, when the base body is relatively massive.

Trajectory/Path Planning. Ullman and Cannon (1989) discussed important issues associated with catching a free-floating object that is initially out of reach of the robot. Trajectory requirements for catching a moving object are described, and a dual-arm two-link planar space manipulator is simulated using a computed torque algorithm. Dubowsky and Torres (1991) employed the Virtual Manipulator Approach in path planning of space manipulators to minimize spacecraft attitude disturbances. Xu (1993) presented a measure of dynamic coupling in free-floating space robotic systems, based on momentum conservation laws. The dynamic coupling factor is defined based on the matrix which relates the end-effector motion and the base body motion, and can be employed in planning robot motions. Nakamura and Mukherjee (1993) presented a trajectory planning scheme that exploits the nonholonomic redundancy of SFFR to avoid joint limits and obstacles. The scheme was developed for a 6-DOF SFFR, and simulation results were included. Yamada et al. (1995) presented a path planning scheme for the single arm of a free-floating satellite which is equipped with momentum wheels. The method utilizes the angular momentum of the base, yet avoids nutation which occurs unless the final satellite attitude is the same as the initial one. Nagamatsu et al. (1996) developed a capture strategy to retrieve a tumbling free-flying object. A simplified dynamics model of the object attitude motion was used to approximate a complex nutation motion by a superposition of rotational motions with constant angular velocities, and the capture planning was introduced based on the proposed model. The transpose Jacobian controller was used for the manipulator control, in both simulation and experimental studies.

Control. Umetani and Yoshida (1989) employing the generalized Jacobian matrix approach, described the differential kinematics of space manipulators. The inverse kinematics problem is solved analytically, and a *resolved motion rate control* is developed to compensate for spacecraft motion. Yoshida et al. (1991) applied this method to the control of a multiple arm system. Alexander and Cannon (1990) developed an algorithm, called the *extended operational-space method* to control the motion of a SFFR, and presented both simulation and experimental results. In this algorithm, the actuator torque vector for the manipulator is calculated based on a reference model, where the spacecraft position and attitude actuators are assumed to be "off" or else to be given and known to the manipulator controller. Fujii et al. (1990) studied dynamics and control of a SFFR with a two-link manipulator in planar motion. The control is performed to make the position and velocity of the end-effector coincide with those of a moving object, in free-floating mode. Papadopoulos and Dubowsky (1991b) suggested that nearly any control algorithm which can be used for fixed-based manipulators can also be employed in the control of free-floating systems provided that the unique dynamical problems of these systems are considered. They have also proposed a model-based algorithm to control the motion of a single arm manipulator in free-flying mode. Yokokohji et al. (1993) studied efficient algorithms for computing the generalized Jacobian matrix, and presented the *resolved acceleration control* for multiple arm space robots. In this algorithm, based on a modified Newton-Euler recursive method, all computations start from the end-effector, so as not to compute the actual acceleration of the spacecraft, also parallel computations of multiple arms becomes possible. Dubowsky and Papadopoulos (1993) focused on the dynamics and control problems unique in rigid space robotic systems, and discussed some of the efforts being done in this field.

Mukherjee and Chen (1993) studied control strategies for changing the configuration of all joints of an underactuated space manipulator. The conditions for controlling only the actuated joints, and all of the system joints, are studied separately. A planar three-link underactuated space manipulator was simulated to demonstrate the application of the obtained results. Agrawal and Desmier (1993) developed mathematical models for different motion primitives in space. Propulsion, collision, catching, and assembly operations were discussed, and some simulation results for a dual-arm space robot in planar motion are presented. Wee and Walker (1993) studied the dynamics of contact between space robots, and proposed an algorithm to achieve both trajectory tracking and impulse minimization.

Yoshida and Nenchev (1995) studied the problem of estimating and minimizing the impulsive reaction force both at the end-effector and at the base. Based on the null-space of the system inertia matrix, they try to find out proper manipulator configurations, to achieve a safe capture and minimize the impact.

Experimental Studies. Carusone et al. (1993) developed a control algorithm to provide accurate end-effector tracking for structurally flexible space manipulators. Instead of linearizing the system equations about the desired trajectory which would result in a time-varying system, a series of steady-state time-invariant models are utilized to reduce computational requirements, and make it easier to handle various trajectories. The algorithm is implemented on a two-link planar manipulator, with the aim of tracking circular and square paths, and the obtained experimental results are compared to those of independent joint PID control implementations. Ejiri et al. (1994) developed a testbed for space robot technologies, and presented some experimental results for a satellite berthing maneuver with a two-armed space robot. Yoshida (1995) presented a summary of theoretical and experimental space robotic research activities, using the Experimental Free-FlOating RoboT Satellite (EFFORTS-I and -II) simulators. The testbed can mechanically simulate the planar floating dynamics of a single or double arm system. Dickson and Cannon (1995) developed The *Decentralized Object Impedance Control*, and presented some experimental results for the capture, transportation, and docking of an object by two free-flying robots in planar motion. The algorithm is an extension of the Object Impedance Control, as discussed in the previous section, to maneuvers with multiple participating robots.

1.3 Structure of the Work

1.3.1 Objectives

Most of the reported studies have focused on the motion control of a single arm manipulator in free-floating mode, i.e. an end-effector moves toward a target in the inertial or spacecraft body-fixed frame with no significant force interactions between the environment and any part of the system. A payload can be considered as a known disturbance added to the last link at the time of capture (Jaar et al. (1992)), while coordination and control of the base and its multiple arms to capture and manipulate space objects has not received much attention. To achieve the goal of capturing and manipulating space objects (which may be passive or include some internal momentum source), this research work focuses on the following issues:

- Kinematics and dynamics modelling of *multiple arm* SFFR;
- Study and development of control strategies applicable in space;
- Motion control of the end-effectors coordinated with the base to chase a moving object according to planned trajectories;
- Trajectory tracking control following object capture, where it may be in contact with its environment;
- Development of a 3-dimensional simulation code for SFFR, in both computational and graphical environments.

1.3.2 Research Tools

Most of the analytical derivations are executed in a symbolic computation environment (MAPLE), without which most of the simulations would not have been possible. The dynamics modelling code has been run for some simple examples, and the results are verified by comparing them with those of hand-calculations. However, since some complicated terms may vanish in the dynamics equations of simple systems, the final model has to be verified in a general problem. This is done by developing an alternative code at a very fundamental level, and comparing the numerical results of both.

Simple cases are simulated in MATLAB, while the simulation code for general SFFR model is in FORTRAN. The veracity of the simulation results have been investigated by comparing the solution for a few simplified examples with solutions available in previous studies, Dubowsky and Papadopoulos (1991a, b). Physical intuition, and investigation of limiting cases were also employed to verify the simulation results. The code has been also used to eliminate programming oversights in the software developed by an independent research group in Japan (Masutani, Y., Osaka University). Using Graphics Library commands, a graphical simulation code for SFFR maneuvers has been developed in C, which demonstrates the results of computational simulations. Running the code on an SGI Indigo 2, with a 4400 processor, yields a smooth animated picture of the maneuver.

1.3.3 Thesis Outline

Two basic approaches for modelling the kinematics of a multi-body space robotic system are developed in Chapter 2. The *barycentric vector approach* is defined based on taking the system center of mass as a representative point for the translational motion, and using a set of the body-fixed vectors which reflect both mass properties and geometric parameters. On the other hand, taking a point on the spacecraft as that representative point for the translational motion (preferably its CM), defines the so-called *direct path approach* which results in more compact equations of motion. In Chapter 3, based on both kinematics approaches, the dynamics modelling of space robotic systems is discussed. The emphasis will be on the direct path approach, to develop a concise explicit dynamics model of multi-manipulator space robots in free-flying mode.

In Chapter 4, appropriate trajectories for the spacecraft and its manipulators motion are planned which lead to capture of moving objects in space. Ensuring smooth operation and reduced disturbances on both the spacecraft and the object just before grasping, these trajectories take into account the target relative motion, and thruster or actuator saturation limits. Then, two model-based control algorithms, based on an Euler angle and an Euler parameter description of the orientation, and a transpose Jacobian (TJ) control algorithm are developed. These algorithms permit control of both the spacecraft and its appendages in their task space. Euler angle model-based control algorithm (MB1) presents the inconvenience of representational singularities, while Euler parameter model-based control algorithm (MB2) overcomes these non-physical singularities. The developed control laws are evaluated using three manipulator or appendages free-flyer examples, in both planar and spatial maneuvers. Comparing the performance of the TJ algorithm to those of different model-based algorithms, illustrates the eligibility of this simple algorithm in controlling highly nonlinear and complex systems, with many Degrees of Freedom (DOF). This result motivates further work on this algorithm, aiming at overcoming the lack of information about the dynamics of the system, a problem which appears more clearly in tracking fast trajectories.

Next, the Modified Transpose Jacobian (MTJ) algorithm is presented in Chapter 5. This new algorithm yields an improved performance over the standard one, by employing stored data of the previous time step control command. The MTJ algorithm is based on an approximation of feedback linearization methods, and does not require a priori knowledge of the plant dynamics terms. Its performance is comparable to that of model-based algorithms, but with a reduced computational burden. Simulation results are presented which compare the performance of the MTJ to that of the TJ and Model-Based algorithms.

To control the system after grasping the object, the new Multiple Impedance Control (MIC) is developed in Chapter 6. This algorithm enforces a controlled impedance of all the manipulator end-points, and of the manipulated object. This guarantees an accordant motion of different parts of the system for performing the task. To reveal the merits of this new algorithm, a simple linear system is considered to present a thorough comparative analysis between the MIC and Object Impedance Control (OIC). Then, application of the MIC law in a system of two cooperating two-link manipulators with an RCC attached to the second end-effector, is simulated. Next, the MIC algorithm is applied in space robotic systems to manipulate space objects. The error analysis shows that under the MIC law, all participating manipulators, the free-flyer base, and the manipulated object exhibit the same impedance behavior.

Chapter 7 reviews the results obtained in this research, conclusions, and some remarks on future work.

1.4 Contributions

Major contributions of this research work are:

- Extension of the Barycentric Vector Approach in space robotics to include multiple arm dynamics, Papadopoulos and Moosavian (1994a);
- Comparison between alternative kinematics/dynamics approaches in space robotics, Papadopoulos and Moosavian (1994b);
- Development of the Modified Transpose Jacobian (MTJ) algorithm, Papadopoulos and Moosavian (1994c);
- Coordination and motion control of multi-manipulator space robots, based on appropriate planned trajectories, resulting in symmetric motion of the manipulators during capture (to minimize spacecraft disturbances), Papadopoulos and Moosavian (1995);
- Development of the Multiple Impedance Control (MIC) and its implementation in space robotic systems;
- Development of a symbolic code based on a concise explicit dynamics model of multi-manipulator space free-flyers, and a 3-dimensional simulation code for SFFR (in both computational and graphical environments).

Chapter 2

Kinematics of Space Free-Flyers with Multiple Manipulators

2.1 Introduction

This chapter studies the kinematics of a multiple manipulator Space Free-Flying Robot, (SFFR). Two basic approaches for kinematics modelling of a rigid multi-body space robotic system are developed. Taking the center of mass of the whole system as a representative point for the system's translational motion, and using a set of body-fixed vectors which reflect both geometric configuration and mass distribution of the system, characterizes the so-called *barycentric vector approach*. This approach results in decoupling the total linear and angular motion from the rest of the equations, when no external forces/torques are applied on the system. On the other hand, taking a point on the base body as the representative point for the system's translational motion, (preferably the center of mass of the base), characterizes the so-called *direct path method*. This approach, eventually, results in a larger number of dynamics equations with simpler terms which have clearer physical meaning. Using the direct path approach seems reasonable when dealing with multiple arm systems, and especially in the presence of external forces/torques.

In Section 2.2, free-flyer kinematics is developed using a minimum set of body-fixed *barycentric vectors*. Position analysis based on the definition of these vectors, and velocity analysis leads to derivation of a system's Jacobian matrix. In Section 2.3, free-flyer kinematics is developed based on the direct path approach, using a set of body-fixed vectors. Discussions of the developed approaches, will be presented in Section 2.4.

2.2 The Barycentric Vector Approach

2.2.1 Frame Assignment and Position Analysis

In this section, using a minimum set of body-fixed *barycentric vectors*, the kinematics of a rigid multiple arm free-flying space robotic system is developed. The motion of the system center of mass (CM) is used to describe system translation with respect to an inertial frame of reference, **XYZ**. The body 0 in Figure 2.1, represents the spacecraft of the free-flyer, which is connected to n manipulators or appendages, each with N_m links. Manipulator joints are revolute and have a single DOF.

The joint angles and rates are represented by $K \times 1$ column vectors $\boldsymbol{\theta} = \left(\boldsymbol{\theta}^{(1)T}, \boldsymbol{\theta}^{(2)T}, \cdots, \boldsymbol{\theta}^{(n)T}\right)^T$, and $\dot{\boldsymbol{\theta}} = \left(\dot{\boldsymbol{\theta}}^{(1)T}, \dot{\boldsymbol{\theta}}^{(2)T}, \cdots, \dot{\boldsymbol{\theta}}^{(n)T}\right)^T$, where $\boldsymbol{\theta}^{(m)}$ is an $N_m \times 1$ column vector which contains the joint angles of the m-th manipulator, and $K = \sum_{m=1}^{n} N_m$. The total degrees-of-freedom (DOF) of the system are $N = K + 6$.

The inertial position of an arbitrary point P, $\mathbf{R}_P$, can be written as

$$\mathbf{R}_P = \mathbf{R}_{CM} + \boldsymbol{\rho}_P \tag{2.1}$$

and

$$\boldsymbol{\rho}_P = \boldsymbol{\rho}_{C_i} + \mathbf{r}_{p/C_i} \tag{2.2}$$

where $\boldsymbol{\rho}_P$ is the position vector of P with respect to the system CM, $\mathbf{R}_{CM}$ is the inertial position of the system CM, C_i is the CM of the i-th body, $\boldsymbol{\rho}_{C_i}$ is its position vector with respect to the system CM, and $\mathbf{r}_{p/C_i}$ is the position vector of P with respect to C_i. Next,

ρ_{C_i} can be computed and expressed in terms of barycentric vectors. Note that, for simplicity, extra subscripts and superscripts are not added in the above equations. When more precise specification is required, subscript "0" is used for the base, and a right superscript corresponding to a specific manipulator, and a subscript referring to a specific body of that manipulator, will be added.

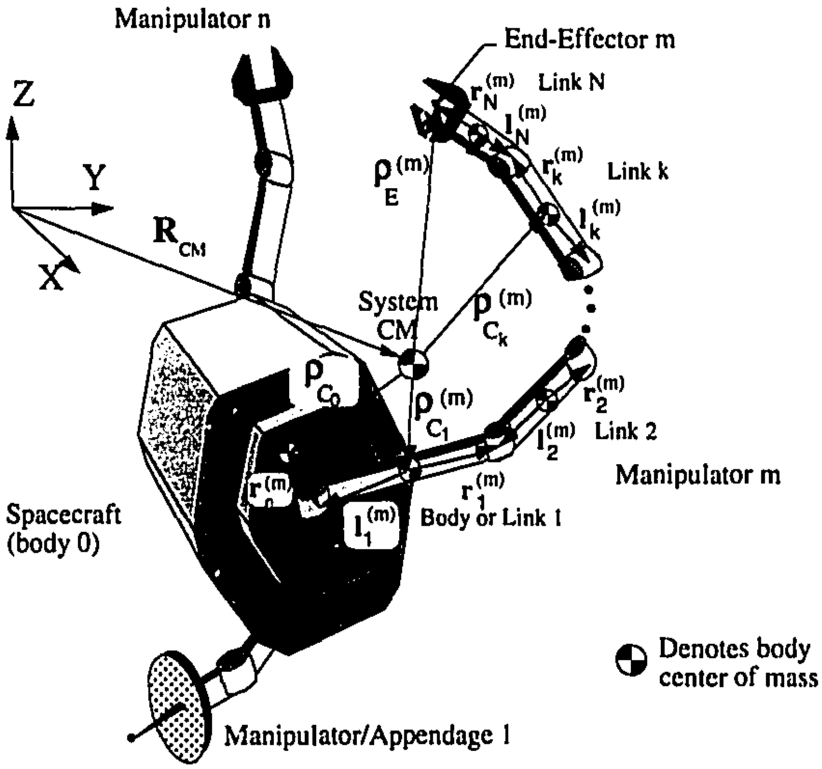


Figure 2.1: A free-flying space robotic system with n manipulators.

2.2.2 Definition of Barycentric Vectors

Vectors ρ_{C_i} in Eq. (2.2), are the position vectors of the CM of the i-th body with respect to the system CM, so they can be computed using

$$m_0 \rho_{C_0} + \sum_{m=1}^{n} \sum_{i=1}^{N_m} m_i^{(m)} \rho_{C_i}^{(m)} = 0 \tag{2.3}$$

and the following geometrical relationships

$$\begin{cases} \boldsymbol{\rho}_{C_i}^{(1)} - \boldsymbol{\rho}_{C_{i-1}}^{(1)} = \mathbf{r}_{i-1}^{(1)} - \mathbf{l}_i^{(1)} & i = 1, \cdots, N_1 \\ \vdots & \\ \boldsymbol{\rho}_{C_i}^{(m)} - \boldsymbol{\rho}_{C_{i-1}}^{(m)} = \mathbf{r}_{i-1}^{(m)} - \mathbf{l}_i^{(m)} & i = 1, \cdots, N_m \\ \vdots & \\ \boldsymbol{\rho}_{C_i}^{(n)} - \boldsymbol{\rho}_{C_{i-1}}^{(n)} = \mathbf{r}_{i-1}^{(n)} - \mathbf{l}_i^{(n)} & i = 1, \cdots, N_n \end{cases} \tag{2.4}$$

where the superscript "m" corresponds to the m-th manipulator, the subscript "i" refers to the i-th body of that manipulator. The system of Eqs. (2.3) and (2.4), represents a system of K+1 vector equations with K+1 unknowns ($\boldsymbol{\rho}_{C_i}$), and can be solved to yield

$$\boldsymbol{\rho}_{C_0} = \tilde{\mathbf{e}}_0 + \sum_{m=1}^{n} \sum_{k=1}^{N_m} \tilde{\mathbf{l}}_k^{(m)} \tag{2.5a}$$

$$\boldsymbol{\rho}_{C_i}^{(m)} = \tilde{\mathbf{r}}_0^{(m)} + \sum_{\substack{j=1 \\ j \neq m}}^{n} \sum_{k=1}^{N_j} \tilde{\mathbf{l}}_k^{(j)} + \sum_{k=1}^{N_m} \tilde{\mathbf{v}}_{ki}^{(m)} \qquad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \tag{2.5b}$$

where ($\tilde{\bullet}$) denotes body-fixed *barycentric vectors* defined as

$$\tilde{\mathbf{v}}_{ki}^{(m)} = \begin{cases} \tilde{\mathbf{r}}_k^{(m)} = \mathbf{r}_k^{(m)} - \mathbf{e}_k^{(m)} & k < i \\ \tilde{\mathbf{e}}_k^{(m)} = -\mathbf{e}_k^{(m)} & k = i \\ \tilde{\mathbf{l}}_k^{(m)} = \mathbf{l}_k^{(m)} - \mathbf{e}_k^{(m)} & k > i \end{cases} \qquad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \tag{2.6}$$

where referring to Figure 2.1, vectors $\mathbf{l}_i^{(m)}$ and $\mathbf{r}_i^{(m)}$ are constant body-fixed vectors which describe the position of joints i and i+1 with respect to C_i, respectively, and $\mathbf{e}_0$ and $\mathbf{e}_i^{(m)}$ are computed as

$$\mathbf{e}_0 = \sum_{m=1}^{n} \mathbf{r}_0^{(m)} \mu_1^{(m)} \tag{2.7a}$$

$$\mathbf{e}_i^{(m)} = \mathbf{l}_i^{(m)}(1 - \mu_i^{(m)}) + \mathbf{r}_i^{(m)} \mu_{i+1}^{(m)} \tag{2.7b}$$

The quantity $\mu_i^{(m)}$ describes the ratio of the outboard mass after the i-th joint of the m-th manipulator with respect to the total mass, and is given by

$$\mu_i^{(m)} = \sum_{k=i}^{N_m} \frac{m_k^{(m)}}{M} \qquad i = 1, \cdots, N_m \quad \text{and} \quad \mu_{N_m+1}^{(m)} = 0 \tag{2.7c}$$

M is the total mass of the system, and $m_k^{(m)}$ is the mass of the k-th body of the m-th manipulator. Considering Eqs. (2.6) and (2.7), it can be seen that barycentric vectors are physically meaningful. For the i-th link of the m-th manipulator, if an augmented body is formed by concentrating the inboard and outboard masses at the corresponding joint of both ends, then $\mathbf{e}_i^{(m)}$ describes the CM position of this augmented body with respect to the real CM of the link. Taking the CM of the augmented body as a reference point, vectors $\tilde{\mathbf{e}}_i^{(m)}$, $\tilde{\mathbf{l}}_i^{(m)}$, and $\tilde{\mathbf{r}}_i^{(m)}$ describe the CM position of the link, position of joints i and i+1 with respect to that point, respectively.

Substitution of Eqs. (2.5a) and (2.5b), for $\boldsymbol{\rho}_{C_i}$, into Eq. (2.2), and the result into Eq. (2.1) completes the position analysis

$$P \in Base: \qquad \mathbf{R}_p^{(0)} = \mathbf{R}_{CM} + \tilde{\mathbf{e}}_0 + \sum_{m=1}^{n}\sum_{k=1}^{N_m} \tilde{\mathbf{l}}_k^{(m)} + \mathbf{r}_{p/C_0} \tag{2.8a}$$

$$P \in Link_i^{(m)}: \qquad \mathbf{R}_{p_i}^{(m)} = \mathbf{R}_{CM} + \tilde{\mathbf{r}}_0^{(m)} + \sum_{\substack{j=1 \\ j \neq m}}^{n}\sum_{k=1}^{N_j} \tilde{\mathbf{l}}_k^{(j)} + \sum_{k=1}^{N_m} \tilde{\mathbf{v}}_{ki}^{(m)} + \mathbf{r}_{p/C_i^{(m)}} \tag{2.8b}$$

Note that the above and the following results are in terms of invariant body-fixed vectors. To obtain scalar equations, appropriate transformation matrices for each term must be employed. It should be mentioned that, based on the spacecraft attitude and corresponding joint angles, orientation of any link of the system can also be obtained.

2.2.3 Velocity Analysis

To obtain the inertial velocity of point P, $\dot{\mathbf{R}}_p$, Eqs. (2.1) and (2.2) are differentiated with respect to time, which results in

$$\dot{\mathbf{R}}_p = \dot{\mathbf{R}}_{CM} + \dot{\boldsymbol{\rho}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{p/C_i} \tag{2.9}$$

where $\dot{\mathbf{R}}_{CM}$ is velocity of the system center of mass, and $\dot{\boldsymbol{\rho}}_{C_i}$ can be obtained by differentiation of Eqs. (2.5a) and (2.5b) which describe $\boldsymbol{\rho}_{C_i}$ in terms of barycentric vectors. Note that the barycentric vectors, according to the definition, are body-fixed vectors with constant length (as long as system mass distribution does not change). Therefore, differentiation of Eqs. (2.5a) and (2.5b) yields

$$\dot{\boldsymbol{\rho}}_{C_0} = \boldsymbol{\omega}_0 \times \tilde{\mathbf{e}}_0 + \sum_{m=1}^{n} \sum_{k=1}^{N_m} \boldsymbol{\omega}_k^{(m)} \times \bar{\mathbf{l}}_k^{(m)} \qquad (2.10a)$$

$$\dot{\boldsymbol{\rho}}_{C_i}^{(m)} = \boldsymbol{\omega}_0 \times \tilde{\mathbf{r}}_0^{(m)} + \sum_{\substack{j=1 \\ j \neq m}}^{n} \sum_{k=1}^{N_j} \boldsymbol{\omega}_k^{(j)} \times \bar{\mathbf{l}}_k^{(j)} + \sum_{k=1}^{N_m} \boldsymbol{\omega}_k^{(m)} \times \tilde{\mathbf{v}}_{ki}^{(m)} \qquad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \qquad (2.10b)$$

where $\boldsymbol{\omega}$'s are angular velocities of individual bodies.

Substitution of Eqs. (2.10a) and (2.10b), for $\dot{\boldsymbol{\rho}}_{C_i}$, into Eq. (2.9) completes the velocity analysis

$P \in$ *Base*: $$\dot{\mathbf{R}}_p^{(0)} = \dot{\mathbf{R}}_{CM} + \boldsymbol{\omega}_0 \times \tilde{\mathbf{e}}_0 + \sum_{m=1}^{n} \sum_{k=1}^{N_m} \boldsymbol{\omega}_k^{(m)} \times \bar{\mathbf{l}}_k^{(m)} + \boldsymbol{\omega}_0 \times \mathbf{r}_{p/C_0} \qquad (2.11a)$$

$P \in$ *Link*$_i^{(m)}$: $$\dot{\mathbf{R}}_{p_i}^{(m)} = \dot{\mathbf{R}}_{CM} + \boldsymbol{\omega}_0 \times \tilde{\mathbf{r}}_0^{(m)} + \sum_{\substack{j=1 \\ j \neq m}}^{n} \sum_{k=1}^{N_j} \boldsymbol{\omega}_k^{(j)} \times \bar{\mathbf{l}}_k^{(j)} + \sum_{k=1}^{N_m} \boldsymbol{\omega}_k^{(m)} \times \tilde{\mathbf{v}}_{ki}^{(m)} + \boldsymbol{\omega}_i^{(m)} \times \mathbf{r}_{p/C_i^{(m)}} \qquad (2.11b)$$

It should be emphasized that in order to perform the foregoing vector sums, all vectors must be expressed in the same coordinate frame.

For single DOF joints, the angular velocity of an individual body can be obtained as

$$\boldsymbol{\omega}_k^{(m)} = \boldsymbol{\omega}_0 + \sum_{i=1}^{k} \dot{\theta}_i^{(m)} \mathbf{z}_i^{(m)} \qquad \begin{cases} m = 1, \cdots, n \\ k = 1, \cdots, N_m \end{cases} \qquad (2.12)$$

where $\mathbf{z}_i^{(m)}$ is a unit vector along the axis of rotation of the i-th joint of the m-th manipulator, and $\dot{\theta}_i^{(m)}$ is the corresponding joint angle rate.

2.2.4 Jacobian Matrix Associated with some Point and Link

Choosing a set of coordinates as system *generalized coordinates*, the linear velocity of an arbitrary point P, and the angular velocity of the corresponding body, can be related to the time derivative of generalized coordinates (i.e. generalized speeds) through a *Jacobian* *matrix*. For instance, if point P belongs to the i-th body of the m-th manipulator, it can be written

$$\begin{Bmatrix} \dot{\mathbf{R}}_p \\ \boldsymbol{\omega}_i^{(m)} \end{Bmatrix} = \mathbf{J}_{i,p}^{(m)} \nu \tag{2.13a}$$

where $\mathbf{J}_{i,p}^{(m)}$ represents a Jacobian matrix, and ν is the vector of generalized speeds, which can be defined as

$$\nu = (\dot{\mathbf{R}}_{CM}^T, \boldsymbol{\omega}_0{}^T, \dot{\boldsymbol{\theta}}^T)^T \tag{2.13b}$$

Then, based on Eqs. (2.11b), and (2.12), $\mathbf{J}_{i,p}^{(m)}$ can be obtained as

$$\mathbf{J}_{i,p}^{(m)} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{J}_1^{(m)} & \mathbf{J}_2^{(m)} \\ \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{J}_3^{(m)} \end{bmatrix}_{6\times N} \tag{2.14}$$

where

$$\mathbf{J}_1^{(m)} = -\left[\mathbf{T}_0 \; {}^0\tilde{\mathbf{r}}_0^{(m)} + \sum_{\substack{j=1 \\ j\neq m}}^{n} \sum_{k=1}^{N_j} \mathbf{T}_k^{(j)} \, {}^k\tilde{\mathbf{l}}_k^{(j)} + \sum_{k=1}^{N_m} \mathbf{T}_k^{(m)} \, {}^k\tilde{\mathbf{v}}_{ki,p}^{(m)} \right]^\times \tag{2.15a}$$

$$\mathbf{J}_2^{(m)} = -\sum_{\substack{j=1 \\ j\neq m}}^{n} \sum_{k=1}^{N_j} [\mathbf{T}_k^{(j)} \, {}^k\tilde{\mathbf{l}}_k^{(j)}]^\times \mathbf{E}_k^{(j)} - \sum_{k=1}^{N_m} [\mathbf{T}_k^{(m)} \, {}^k\tilde{\mathbf{v}}_{ki,p}^{(m)}]^\times \mathbf{E}_k^{(m)} \tag{2.15b}$$

$$\mathbf{J}_3^{(m)} = \mathbf{E}_i^{(m)} \tag{2.15c}$$

$\mathbf{T}_0$ and $\mathbf{T}_j^{(k)}$ are rotation matrices between body-fixed frames and the inertial frame, while $[\bullet]^\times$ is the *cross product operator*, and

$$\tilde{\mathbf{v}}_{jl,p}^{(k)} = \tilde{\mathbf{v}}_{jl}^{(k)} + \delta_{jl} \, \mathbf{r}_{p/C_l^{(m)}} \tag{2.15d}$$

$$\mathbf{E}_j^{(k)} = \begin{bmatrix} \mathbf{0}_{3\times b} & \mathbf{T}_1^{(k)} \, {}^1\mathbf{z}_1^{(k)} & \cdots & \mathbf{T}_j^{(k)} \, {}^j\mathbf{z}_j^{(k)} & \mathbf{0} \end{bmatrix}_{3\times K} \tag{2.15e}$$

where δ_{jl} is *Kronecker delta*, $b = \sum_{l=1}^{k-1} \mathrm{N}_l$, and ${}^j\mathbf{z}_j^{(k)} \equiv (0,0,1)^T$ is a unit vector along the axis of rotation of the j-th joint of the k-th manipulator expressed in its own body-fixed frame. Note that a left superscript refers to the frame in which the corresponding vector is expressed, and it disappears for the inertial frame.

Similarly, based on Eqs. (2.11a), and (2.12), $\mathbf{J}_{0,p}$ can be obtained for a point P on the spacecraft as

$$\mathbf{J}_{0,p} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{J}_1^{(0)} & \mathbf{J}_2^{(0)} \\ \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0} \end{bmatrix}_{6\times N} \tag{2.16}$$

where

$$\mathbf{J}_1^{(0)} = -\left[\mathbf{T}_0(\,{}^0\tilde{\mathbf{e}}_0 + \mathbf{r}_{p/c_0}) + \sum_{m=1}^{n} \sum_{k=1}^{N_j} \mathbf{T}_k^{(m)}\,{}^k\tilde{\mathbf{l}}_k^{(m)}\right]^{\times} \tag{2.17a}$$

$$\mathbf{J}_2^{(0)} = -\sum_{m=1}^{n} \sum_{k=1}^{N_j} [\mathbf{T}_k^{(m)}\,{}^k\tilde{\mathbf{l}}_k^{(m)}]^{\times}\,\mathbf{E}_k^{(m)} \tag{2.17b}$$

Taking the whole system CM as a representative point for the system's translation, and using a set of body-fixed barycentric vectors, the kinematics of a rigid multiple arm SFFR was developed. Next, the spacecraft CM is taken as the representative point for the system's translational motion, and the kinematics of a SFFR is developed in terms of body-fixed vectors.

2.3 The Direct Path Method

2.3.1 Frame Assignment and Position Analysis

In this section, using a set of body-fixed geometric vectors, the kinematics of a rigid multiple arm free-flying space robotic system is developed. The motion of the spacecraft center of mass (CM) is used to describe the system global translation with respect to an inertial frame of reference, **XYZ**. The rest of the definitions described in Section 2.2.1, are applicable here to the same extent as before.

Considering Figure 2.2, the inertial position of an arbitrary point P, $\mathbf{R}_p$, can be written as

$$\mathbf{R}_p = \mathbf{R}_{C_0} + \mathbf{r}_p \tag{2.18}$$

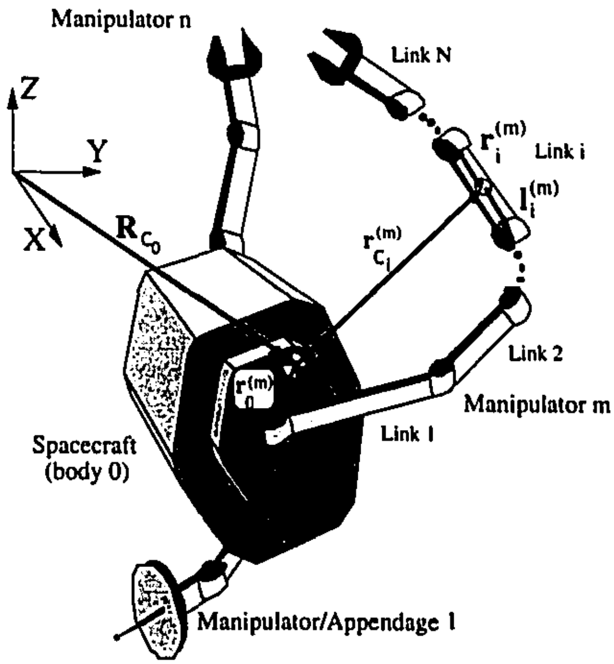


Figure 2.2: A free-flying space robotic system with n manipulators.

and

$$\mathbf{r}_P = \mathbf{r}_{C_i} + \mathbf{r}_{p/C_i} \tag{2.19}$$

where $\mathbf{R}_{C_0}$ is the inertial position of the spacecraft CM, $\mathbf{r}_p$ is the position vector of point P with respect to the spacecraft CM, and $\mathbf{r}_{C_i}$ is the CM position vector of the i-th body with respect to the spacecraft CM. Referring to Figure 2.2, $\mathbf{r}_{C_i}$ can be expressed as follows

$$\mathbf{r}_{C_0} = \mathbf{0} \tag{2.20a}$$

$$\mathbf{r}_{C_i}^{(m)} = \mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1}(\mathbf{r}_k^{(m)} - \mathbf{l}_k^{(m)}) - \mathbf{l}_i^{(m)} \qquad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \tag{2.20b}$$

where, as before, vectors $\mathbf{l}_i^{(m)}$ and $\mathbf{r}_i^{(m)}$ are body-fixed vectors which describe the position of joints i and i+1 with respect to C_i, see Figure 2.2.

Substitution of Eqs. (2.20a) and (2.20b) for $\mathbf{r}_{C_i}$, into Eq. (2.19), and the result into Eq. (2.18) completes the position analysis and yields

$$P \in Base: \qquad \mathbf{R}_p^{(0)} = \mathbf{R}_{C_0} + \mathbf{r}_{p/C_0} \tag{2.21a}$$

$$P \in Link_i^{(m)}: \quad \mathbf{R}_{p_i}^{(m)} = \mathbf{R}_{C_0} + \mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1}(\mathbf{r}_k^{(m)} - \mathbf{l}_k^{(m)}) - \mathbf{l}_i^{(m)} + \mathbf{r}_{p/C_i^{(m)}} \tag{2.21b}$$

2.3.2 Velocity Analysis

To obtain the inertial velocity of point P, Eq. (2.18) is differentiated, after substituting Eq. (2.19), to yield

$$\dot{\mathbf{R}}_p = \dot{\mathbf{R}}_{C_0} + \dot{\mathbf{r}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{p/C_i} \tag{2.22}$$

where $\dot{\mathbf{R}}_{C_0}$ is velocity of the spacecraft CM. Differentiation of Eqs. (2.20) yields

$$\dot{\mathbf{r}}_{C_0} = \mathbf{0} \tag{2.23a}$$

$$\dot{\mathbf{r}}_{C_i}^{(m)} = \boldsymbol{\omega}_0 \times \mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1} \boldsymbol{\omega}_k^{(m)} \times (\mathbf{r}_k^{(m)} - \mathbf{l}_k^{(m)}) - \boldsymbol{\omega}_i^{(m)} \times \mathbf{l}_i^{(m)} \qquad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \tag{2.23b}$$

where $\boldsymbol{\omega}$'s are angular velocities of individual bodies.

Substitution of Eqs. (2.23a) and (2.23b), for $\dot{\mathbf{r}}_{C_i}$, into Eq. (2.22) completes the velocity analysis

$P \in Base$:
$$\dot{\mathbf{R}}_p^{(0)} = \dot{\mathbf{R}}_{C_0} + \boldsymbol{\omega}_0 \times \mathbf{r}_{p/C_0} \tag{2.24a}$$

$P \in Link_i^{(m)}$:
$$\dot{\mathbf{R}}_{p_i}^{(m)} = \dot{\mathbf{R}}_{C_0} + \boldsymbol{\omega}_0 \times \mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1} \boldsymbol{\omega}_k^{(m)} \times (\mathbf{r}_k^{(m)} - \mathbf{l}_k^{(m)}) - \boldsymbol{\omega}_i^{(m)} \times (\mathbf{l}_i^{(m)} - \mathbf{r}_{p/C_i^{(m)}}) \tag{2.24b}$$

It should be noted that the angular velocity of any individual body, for single DOF joints, can be obtained as defined in Eq. (2.12).

2.3.3 Jacobian Matrix Associated with some Point and Link

The linear velocity of an arbitrary point P on the i-th body of the m-th manipulator, and angular velocity of the corresponding body, can be expressed as

$$\begin{Bmatrix} \dot{\mathbf{R}}_p \\ \boldsymbol{\omega}_i^{(m)} \end{Bmatrix} = \mathbf{J}_{i,p}^{(m)} \nu \tag{2.25a}$$

where $\mathbf{J}_{i,p}^{(m)}$ represents a Jacobian matrix, and ν is the vector of generalized speeds, which is defined as

$$\nu = (\dot{\mathbf{R}}_{C_0}^T, \boldsymbol{\omega}_0^T, \dot{\boldsymbol{\theta}}^T)^T \tag{2.25b}$$

Note that the generalized speeds include $\dot{\mathbf{R}}_{C_0}$ instead of $\dot{\mathbf{R}}_{CM}$, see Eq. (2.13b). Then, based on Eqs. (2.12) and (2.24b), $\mathbf{J}_{i,p}^{(m)}$ can be computed as

$$\mathbf{J}_{i,p}^{(m)} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{J}_1^{(m)} & \mathbf{J}_2^{(m)} \\ \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{J}_3^{(m)} \end{bmatrix}_{6\times N} \tag{2.26}$$

where

$$\mathbf{J}_1^{(m)} = -\left[\mathbf{T}_0\ {}^0\mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1}\left[\mathbf{T}_k^{(m)}({}^k\mathbf{r}_k^{(m)} - {}^k\mathbf{l}_k^{(m)})\right] - \mathbf{T}_i^{(m)}({}^i\mathbf{l}_i^{(m)} - {}^i\mathbf{r}_{p/C_i^{(m)}})\right]^{\times} \tag{2.27a}$$

$$\mathbf{J}_2^{(m)} = -\sum_{k=1}^{i-1}\left[\mathbf{T}_k^{(m)}({}^k\mathbf{r}_k^{(m)} - {}^k\mathbf{l}_k^{(m)})\right]^{\times}\mathbf{E}_k^{(m)} + \left[\mathbf{T}_i^{(m)}({}^i\mathbf{l}_i^{(m)} - {}^i\mathbf{r}_{p/C_i^{(m)}})\right]^{\times}\mathbf{E}_i^{(m)} \tag{2.27b}$$

$$\mathbf{J}_3^{(m)} = \mathbf{E}_i^{(m)} \tag{2.27c}$$

and the definitions given for different terms in Eqs. (2.15), are applicable here, too.

Similar to the above, $\mathbf{J}_{0,p}$ can be obtained for the one corresponding to point P on the spacecraft, based on Eqs. (2.12), and (2.24a)

$$\mathbf{J}_{0,p} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{J}_1^{(0)} & \mathbf{0} \\ \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0} \end{bmatrix}_{6\times N} \tag{2.28}$$

where

$$\mathbf{J}_1^{(0)} = -\left[\mathbf{T}_0\ {}^0\mathbf{r}_{p/C_0})\right]^{\times} \tag{2.29}$$

2.4 Discussion and Conclusions

In this section the two approaches developed for kinematics analysis of SFFR with rigid multiple manipulators, are compared. As revealed by the above formulations, the barycentric vector approach is developed based on

- Taking the center of mass of the whole system as a representative point for the system's translational motion;
- Using a set of body-fixed barycentric vectors which reflect both the geometric configuration and the mass distribution of the system.

On the other hand, the direct path approach is developed based on

- Taking a point on the base body (preferably its CM) as the representative point for the system's translation;
- Using a set of body-fixed geometric vectors.

Comparing the obtained results for position analysis, Eqs. (2.8) compared to Eqs. (2.21), it can be seen that the direct path approach results in single summations and yields more compact relationships. Note that presence of double summations in Eqs. (2.8) means that all system links are contributing in defining the position of any arbitrary point P. This is due to the fact that by taking the center of mass of the whole system as a representative point for the system's translation, the mass distribution over the entire system (represented in Eq. (2.3)) has to be taken into account in writing position relationships.

The difference between the two approaches is more considerable for the velocities, Eqs. (2.11) compared to Eqs. (2.24), because each vector has to be multiplied with the angular velocity of the corresponding body. This leads to a big difference between the resulting Jacobian matrices, Eqs. (2.15) compared to Eqs. (2.27) or Eqs. (2.16, 17) compared to Eqs. (2.28, 29). Note that the complexity of the Jacobian matrix is important because many control algorithms require its computation; these algorithms can be implemented more easily using the direct path approach.

It should be mentioned that the barycentric vector approach is an approach which considers the next step of using kinematics equations in dynamics. In fact, it results in decoupling the total linear and angular motion from the rest of the equations, when no external forces and torques are applied on the system. But, according to the above discussion, the direct path approach results in more compact equations in kinematics and consequently in dynamics. Therefore, using this approach seems reasonable when dealing with multiple arm systems, especially in the presence of external forces and torques. This is to be investigated in the next chapter.

Chapter 3

Dynamics of Space Free-Flyers with Multiple Manipulators

3.1 Introduction

This chapter studies the dynamics of a multiple manipulator Space Free-Flying Robot (SFFR) with rigid links. To apply the general Lagrangian formulation, first the system kinetic energy is derived based on the two alternative kinematics approaches developed in Chapter 2. The obtained results are compared, and it is shown that the *direct path approach* yields more compact expressions. Next, the derivation of the equations of motion is pursued on the basis of using this approach. Explicit derivations of a system's mass matrix, and of the vectors of nonlinear velocity terms, and generalized forces are introduced. The results are summarized in an explicit dynamics model of multiple manipulator SFFR, which can be implemented either *numerically* or *symbolically*. Here, the latter approach is followed, and the developed symbolic code for dynamics modelling, and its verification procedure are described.

In Section 3.3, issues of dynamics relevant to the development of control algorithms, are briefly discussed. First, a *quasi-coordinate formulation* for system dynamics is presented which is useful in developing control algorithms. In this formulation the angular velocity of the spacecraft, instead of the corresponding Euler rates, is chosen and included in the vector of generalized speeds. The system dynamics is also formulated on the basis of choosing *Euler parameters* for orientation representation. This selection introduces algebraic constraints to the system, and the *Natural Orthogonal Complement Method* is applied to obtain independent system of equations of motion. Some specific characteristics of space robotic systems compared to fixed-base manipulators are pointed out at the end of this section. Section 3.4 describes the developed symbolic code for dynamics modelling, and the verification procedure.

3.2 General Lagrangian Formulation

Since a typical maneuver of SFFR is of relatively short length and duration, microgravity and dynamical effects due to orbital mechanics are negligible, compared to control forces. Therefore, the motion of the system is considered with respect to an in-orbit inertial frame of reference (**XYZ**), and the system potential energy is taken equal to zero. So, the general Lagrangian formulation for such system can be written as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right)-\left(\frac{\partial T}{\partial q_i}\right)=Q_i \qquad i=1,\cdots,N \tag{3.1}$$

where T is the system kinetic energy, N is the system degrees-of-freedom, q_i, $\dot{q}_i$, and Q_i are the i-th element of the vector of generalized coordinates, generalized speeds, and generalized forces, respectively. To apply Eq. (3.1), and obtain dynamics equations, first the system kinetic energy, T, has to be derived.

3.2.1 Kinetic Energy Calculations

The system kinetic energy can be written as

$$T=\frac{1}{2}\int_M \dot{\mathbf{R}}_P \cdot \dot{\mathbf{R}}_P \, dM \tag{3.2}$$

where M defines the system's distributed mass, and $\dot{\mathbf{R}}_P$ is velocity of an arbitrary point P. The above expression is now evaluated based on the two different kinematics approaches for multiple manipulator SFFR with rigid elements, developed in the previous chapter.

3.2.1.1 Analysis Based on Barycentric Vector Kinematics

Substitution of Eq. (2.9) for $\dot{\mathbf{R}}_P$ into Eq. (3.2) yields

$$T = \frac{1}{2}\int_M (\dot{\mathbf{R}}_{CM} + \dot{\boldsymbol{\rho}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{p/C_i}) \cdot (\dot{\mathbf{R}}_{CM} + \dot{\boldsymbol{\rho}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{p/C_i})\, dM \tag{3.3}$$

Vectors $\boldsymbol{\rho}_{C_i}$ are written with respect to the system center of mass, therefore

$$\int_M \left(\sum \boldsymbol{\rho}_{C_i}\right) dM = \mathbf{0} \tag{3.4}$$

Using Eqs. (3.4), and further simplifications of Eq. (3.3) lead to

$$T = T_0 + T_1 \tag{3.5a}$$

where

$$T_0 = \frac{1}{2} M\, (\dot{\mathbf{R}}_{CM} \cdot \dot{\mathbf{R}}_{CM}) \tag{3.5b}$$

$$T_1 = \frac{1}{2}\left\{ m_0\, \dot{\boldsymbol{\rho}}_{C_0} \cdot \dot{\boldsymbol{\rho}}_{C_0} + \boldsymbol{\omega}_0 \cdot \mathbf{I}_0 \cdot \boldsymbol{\omega}_0 + \sum_{m=1}^{n} \sum_{i=1}^{N_m} (m_i^{(m)}\, \dot{\boldsymbol{\rho}}_{C_i}^{(m)} \cdot \dot{\boldsymbol{\rho}}_{C_i}^{(m)} + \boldsymbol{\omega}_i^{(m)} \cdot \mathbf{I}_i^{(m)} \cdot \boldsymbol{\omega}_i^{(m)}) \right\} \tag{3.5c}$$

m_0 and $\mathbf{I}_0$ are the mass and inertia dyad of the base with respect to its CM, respectively, and $m_i^{(m)}$ and $\mathbf{I}_i^{(m)}$ are those of the i-th body of the m-th manipulator with respect to its CM. To obtain a detailed expression for T, vectors $\dot{\boldsymbol{\rho}}_{C_0}$ and $\dot{\boldsymbol{\rho}}_{C_i}^{(m)}$ have to be substituted into these equations from Eqs. (2.10) (repeated here)

$$\dot{\boldsymbol{\rho}}_{C_0} = \boldsymbol{\omega}_0 \times \tilde{\mathbf{e}}_0 + \sum_{m=1}^{n} \sum_{k=1}^{N_m} \boldsymbol{\omega}_k^{(m)} \times \tilde{\mathbf{l}}_k^{(m)} \tag{2.10a}$$

$$\dot{\boldsymbol{\rho}}_{C_i}^{(m)} = \boldsymbol{\omega}_0 \times \tilde{\mathbf{r}}_0^{(m)} + \sum_{\substack{j=1 \\ j \neq m}}^{n} \sum_{k=1}^{N_j} \boldsymbol{\omega}_k^{(j)} \times \tilde{\mathbf{l}}_k^{(j)} + \sum_{k=1}^{N_m} \boldsymbol{\omega}_k^{(m)} \times \tilde{\mathbf{v}}_{ki}^{(m)} \qquad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \tag{2.10b}$$

The vector of generalized coordinates is chosen as

$$\mathbf{q} = (\mathbf{R}_{CM}^T, \boldsymbol{\delta}_0^T, \boldsymbol{\theta}^T)^T \tag{3.6}$$

where $\boldsymbol{\delta}_0$ is a set of Euler angles that describe the orientation of the spacecraft. The spacecraft angular velocity can be expressed in terms of the Euler rates as

$${}^0\boldsymbol{\omega}_0 = \mathbf{S}_0(\boldsymbol{\delta}_0)\dot{\boldsymbol{\delta}}_0 \tag{3.7}$$

where $\mathbf{S}_0(\boldsymbol{\delta}_0)$ is a 3×3 matrix, see Meirovitch (1970). The vector ${}^0\boldsymbol{\omega}_0$ is the spacecraft angular velocity expressed in its frame of reference. Therefore, the system kinetic energy can be obtained as

$$T = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{H}(\boldsymbol{\delta}_0, \boldsymbol{\theta})\,\dot{\mathbf{q}} \tag{3.8}$$

where $\mathbf{H}$ is an N×N positive definite mass matrix of the system. Note that $\mathbf{H}$ is a function of the spacecraft attitude and joint angles ($\boldsymbol{\delta}_0$, $\boldsymbol{\theta}$), and is independent from the CM position ($\mathbf{R}_{CM}$).

3.2.1.2 Analysis Based on Direct Path Kinematics

Substitution of Eq. (2.22) for $\dot{\mathbf{R}}_P$ into Eq. (3.2) yields

$$T = \frac{1}{2}\int_M (\dot{\mathbf{R}}_{C_0} + \dot{\mathbf{r}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{p/C_i}) \cdot (\dot{\mathbf{R}}_{C_0} + \dot{\mathbf{r}}_{C_i} + \boldsymbol{\omega}_i \times \mathbf{r}_{p/C_i})\, dM \tag{3.9}$$

which can be simplified, to obtain

$$T = T_0 + T_1 + T_2 \tag{3.10a}$$

where

$$T_0 = \frac{1}{2}M\,(\dot{\mathbf{R}}_{C_0} \cdot \dot{\mathbf{R}}_{C_0}) \tag{3.10b}$$

$$T_1 = \frac{1}{2}\left\{\boldsymbol{\omega}_0 \cdot \mathbf{I}_0 \cdot \boldsymbol{\omega}_0 + \sum_{m=1}^{n}\sum_{i=1}^{N_m}(m_i^{(m)}\,\dot{\mathbf{r}}_{C_i}^{(m)} \cdot \dot{\mathbf{r}}_{C_i}^{(m)} + \boldsymbol{\omega}_i^{(m)} \cdot \mathbf{I}_i^{(m)} \cdot \boldsymbol{\omega}_i^{(m)})\right\} \tag{3.10c}$$

$$T_2 = \dot{\mathbf{R}}_{C_0} \cdot \left(\sum_{m=1}^{n}\sum_{i=1}^{N_m} m_i^{(m)}\,\dot{\mathbf{r}}_{C_i}^{(m)}\right) \tag{3.10d}$$

and $\dot{\mathbf{r}}_{C_i}^{(m)}$ can be substituted from Eq. (2.23b) (repeated here for convenience)

$$\dot{\mathbf{r}}_{C_i}^{(m)} = \boldsymbol{\omega}_0 \times \mathbf{r}_0^{(m)} + \sum_{k=1}^{i-1} \boldsymbol{\omega}_k^{(m)} \times (\mathbf{r}_k^{(m)} - \mathbf{l}_k^{(m)}) - \boldsymbol{\omega}_i^{(m)} \times \mathbf{l}_i^{(m)} \qquad \begin{cases} m = 1, \cdots, n \\ i = 1, \cdots, N_m \end{cases} \tag{2.23b}$$

The vector of generalized coordinates is chosen here as

$$\mathbf{q} = (\mathbf{R}_{C_0}^T, \boldsymbol{\delta}_0^T, \boldsymbol{\theta}^T)^T \tag{3.11}$$

and the system kinetic energy can be written as expressed in Eq. (3.8). Note that in both formulations, expressions for T are in terms of invariant body-fixed vectors. To do the required differentiations in Eq. (3.1), appropriate transformation matrices for each term must be employed.

Next, the obtained expressions for the system kinetic energy, based on the two kinematics approaches, are compared and discussed.

3.2.1.3 Comparison Between the Obtained Results

Considering Eq. (3.1), it can be seen that substitution of Eqs. (3.5), i.e. the system kinetic energy based on barycentric vector kinematics, results in decoupling of the first three equations from the rest of the dynamics equations if no external forces are applied on the system. In fact, the first three equations of motion will be obtained as

$$M\, \ddot{\mathbf{R}}_{CM} = \mathbf{0} \tag{3.12}$$

where $\ddot{\mathbf{R}}_{CM}$ is the system CM acceleration. However, substitution of the system kinetic energy obtained based on the direct path kinematics, i.e. Eqs. (3.10), into Eq. (3.1) does not yield such a decoupling in dynamics equations. This is due to the presence of an additional term, T_2, in the system kinetic energy expression. In fact, differentiation of T_2 with respect to $\dot{\mathbf{R}}_{C_0}$ yields

$$\frac{\partial T_2}{\partial \dot{\mathbf{R}}_{C_0}} = \sum_{m=1}^{n} \sum_{i=1}^{N_m} m_i^{(m)}\, \dot{\mathbf{r}}_{C_i}^{(m)} \tag{3.13}$$

where $\dot{\mathbf{r}}_{C_i}^{(m)}$ is a function of the m-th manipulator joint rates and spacecraft Euler rates, as Eq. (2.23b) shows. Therefore, $\partial T_2 / \partial \dot{\mathbf{R}}_{C_0}$ is a function of all joint rates and spacecraft Euler rates, and its differentiation with respect to time yields a coupled system of dynamics equations. Obviously, this difference can be observed in the first three rows of mass matrix **H** for each case. Similar comments apply to the next three rows, corresponding to the spacecraft Euler rates. The difference in subsequent block which corresponds to the joint rates is investigated, next.

As Eqs. (2.10) and (2.12) reveal, vectors $\dot{\boldsymbol{\rho}}_{C_0}$ and $\dot{\boldsymbol{\rho}}_{C_i}^{(m)}$ are functions of *all* joint rates and spacecraft Euler rates. Therefore, differentiation of the terms $\left(\dot{\boldsymbol{\rho}}_{C_0} \cdot \dot{\boldsymbol{\rho}}_{C_0}\right)$ and $\left(\dot{\boldsymbol{\rho}}_{C_i}^{(m)} \cdot \dot{\boldsymbol{\rho}}_{C_i}^{(m)}\right)$ in Eq. (3.5c) with respect to any joint rate results in a lengthy expression which is a function of all joint rates and spacecraft Euler rates. Subsequent differentiation of the obtained expression with respect to time, as required in the calculation of $d(\partial T / \partial \dot{q}_i) / dt$, yields a lengthy expression, function of second rate of all joint angles and spacecraft Euler angles. This means that the block of the mass matrix **H** which corresponds to the joint rates, if developed based on barycentric vector kinematics, is fully occupied by elements with many terms. On the other hand, considering the direct path approach, vectors $\dot{\mathbf{r}}_{C_i}^{(m)}$ are functions of just a *subset* of joint rates (those of the m-th manipulator) and spacecraft Euler rates, see Eq. (2.23b). Therefore, differentiation of $\left(\dot{\mathbf{r}}_{C_i}^{(m)} \cdot \dot{\mathbf{r}}_{C_i}^{(m)}\right)$ with respect to any joint rate out of the corresponding subset is zero. Consequently, if developed based on direct path kinematics, the block of mass matrix **H** which corresponds to the joint rates is occupied by elements with fewer terms. It should be noted that for a multiple manipulator SFFR, this block is most likely the main part of mass matrix.

So far, the main concern was the first term in Eq. (3.1), $d(\partial T / \partial \dot{q}_i) / dt$, and the difference between the obtained mass matrices as a consequence of dealing with this term. The other term in this equation, $\partial T / \partial q_i$, which results in the vector of nonlinear velocity terms, should also be taken into consideration. Following a similar discussion, it can be shown that a significant difference will appear between the two approaches in calculating

$\partial T / \partial q_i$, and the direct path kinematics results in a vector of nonlinear velocity with more concise terms.

Based on the above discussion, it can be concluded that barycentric vector kinematics may result in lengthy dynamics equations specially for multiple manipulator SFFR, while direct path kinematics results in relatively compact dynamics equations. This is a vital difference which is of particular importance in the execution time of simulation routines. Furthermore, the main advantage of the barycentric approach, i.e. being able to decouple the total linear and angular motion from the rest of the equations (if no external forces/torques are applied on the system), is not a substantial concern for this research work[2]. Therefore, in the next section the focus is on the direct path kinematics to develop an explicit dynamics model of a multiple manipulator SFFR.

3.2.2 Equations of Motion via the Direct Path Approach

Applying the general Lagrangian formulation, Eq. (3.1), where the system kinetic energy is substituted from Eq. (3.10), the equations of motion can be obtained as

$$\mathbf{H}(\boldsymbol{\delta}_0, \boldsymbol{\theta})\ddot{\mathbf{q}} + \mathbf{C}(\boldsymbol{\delta}_0, \dot{\boldsymbol{\delta}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{Q}(\boldsymbol{\delta}_0, \boldsymbol{\theta}) \tag{3.14}$$

where the vector of generalized coordinates $\mathbf{q}$ has been already defined in Eq. (3.11), $\mathbf{C}$ is an N×1 vector which contains all the nonlinear velocity terms (in a microgravity environment), and $\mathbf{Q}$ is the N×1 vector of generalized forces given by

$$\mathbf{Q} = \begin{Bmatrix} \mathbf{0}_{6\times1} \\ \boldsymbol{\tau}_{K\times1} \end{Bmatrix} + \sum_{p=1}^{i_f} \mathbf{J}_{0,p}{}^T \mathbf{F}_{0,p} + \sum_{m=1}^{n} \sum_{i=1}^{N_m} \sum_{p=1}^{i_f} \mathbf{J}_{i,p}^{(m)\,T} \mathbf{F}_{i,p}^{(m)} \tag{3.15}$$

in which $\mathbf{F}_{0,p}$ is the p-th external force/moment applied on the spacecraft, $\mathbf{F}_{i,p}^{(m)}$ is the p-th external force/moment applied on the i-th body of the m-th manipulator, i_f is the number

---

2- It should be noted that to develop model-based algorithms for controlling a *free-floating system*, the dynamics model obtained based on the direct path kinematics, has to be reduced by mathematical techniques such as *Orthogonal Complement Method*. However, the dynamics model obtained in terms of barycentric vectors, can be directly reduced and employed for such a purpose.

of applied forces/moments on the corresponding body, and $\mathbf{J}_{l,p}^{(m)}$ is a Jacobian matrix corresponding to the point of force/moment application. Note that Eq. (3.15) can be obtained based on the definition of generalized forces. This equation can be rearranged, so that actuator forces/torques are displayed explicitly. For instance, if all external forces except the ones applied on the spacecraft are zero, $\mathbf{Q}$ can be written as

$$\mathbf{Q} = \mathbf{J}_Q \begin{Bmatrix} {}^0\mathbf{f}_s \\ {}^0\mathbf{n}_s \\ \boldsymbol{\tau}_{K\times 1} \end{Bmatrix} \tag{3.16}$$

where ${}^0\mathbf{f}_s$ and ${}^0\mathbf{n}_s$ are the net force and moment applied on the spacecraft, and $\mathbf{J}_Q$ is an N×N Jacobian matrix. For a well designed system, $\mathbf{J}_Q$ remains nonsingular, i.e. any required $\mathbf{Q}$ can be produced by the system's actuators.

Next, to obtain an *explicit* dynamics model of multiple manipulator SFFR, mathematical analyses are presented to help in calculating the mass matrix, the vector of nonlinear velocity terms, and the generalized forces.

3.2.2.1 Preliminary Derivations

The system kinetic energy (as expressed in Eq. (3.10)) regardless of body specifications, is composed of three typical terms

$$a_1 = \frac{1}{2} m\, \dot{\mathbf{r}} \cdot \dot{\mathbf{r}} \tag{3.17a}$$

$$a_2 = \frac{1}{2} \boldsymbol{\omega} \cdot \mathbf{I} \cdot \boldsymbol{\omega} \tag{3.17b}$$

$$a_3 = \dot{\mathbf{R}}_{C_0} \cdot \sum_k m_k \dot{\mathbf{r}}_k \tag{3.17c}$$

So, to differentiate the system kinetic energy according to Eq. (3.1), such terms have to be differentiated. Therefore, preliminary calculations in differentiation of these terms are presented in this section, resulting in three *formats* which describe the contribution of each term to the equations of motion. These formats, obtained in Appendix A, will be used in deriving the system dynamics model in the following sections.

Considering the first term, Eq. (3.17a), it is obtained that

$$\frac{d}{dt}\left(\frac{\partial a_1}{\partial \dot{q}_i}\right)-\frac{\partial a_1}{\partial q_i}=\left[m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\frac{\partial \mathbf{r}}{\partial q_1} \quad \cdots \quad m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\frac{\partial \mathbf{r}}{\partial q_N}\right]\ddot{\mathbf{q}}+ \\ \left[m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}}{\partial q_s \partial q_1}\dot{q}_s\right) \quad \cdots \quad m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}}{\partial q_s \partial q_N}\dot{q}_s\right)\right]\dot{\mathbf{q}} \tag{3.18}$$

which describes *format-I*, defined as contribution of the first typical term to the equations of motion. Note that $\mathbf{r}$ has to be differentiated in the inertial frame (see Appendix A).

Considering the second term, Eq. (3.17b), it can be obtained that

$$\frac{d}{dt}\left(\frac{\partial a_2}{\partial \dot{q}_i}\right)-\frac{\partial a_2}{\partial q_i}=\left[\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_1} \quad \cdots \quad \frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_N}\right]\ddot{\mathbf{q}}+ \\ \left[\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial q_1}+\boldsymbol{\omega}\cdot\mathbf{I}\cdot\frac{\partial^2 \boldsymbol{\omega}}{\partial \dot{q}_i \partial q_1} \quad \cdots \quad \frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial q_N}+\boldsymbol{\omega}\cdot\mathbf{I}\cdot\frac{\partial^2 \boldsymbol{\omega}}{\partial \dot{q}_i \partial q_N}\right]\dot{\mathbf{q}}-\boldsymbol{\omega}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial q_i} \tag{3.19}$$

which describes *format-II*. Note that $\boldsymbol{\omega}$ is differentiated in the body frame (see Appendix A). This will be emphasized by using a left superscript on partial derivatives of $\boldsymbol{\omega}$ in the following formulations, consistent with the notations used by Kane and Levinson (1985).

Finally, considering Eq. (3.17c), it is obtained that

$$\frac{d}{dt}\left(\frac{\partial a_3}{\partial \dot{q}_i}\right)-\frac{\partial a_3}{\partial q_i}=\left[\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k\frac{\partial \mathbf{r}_k}{\partial q_1} \quad \cdots \quad \frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k\frac{\partial \mathbf{r}_k}{\partial q_N}\right]\ddot{\mathbf{q}}+ \\ \left[\frac{\partial \mathbf{R}_{C_0}}{\partial q_1}\cdot\sum_k m_k\frac{\partial \mathbf{r}_k}{\partial q_i} \quad \cdots \quad \frac{\partial \mathbf{R}_{C_0}}{\partial q_N}\cdot\sum_k m_k\frac{\partial \mathbf{r}_k}{\partial q_i}\right]\ddot{\mathbf{q}}+ \\ \left[\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}_k}{\partial q_1 \partial q_s}\dot{q}_s\right) \quad \cdots \quad \frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}_k}{\partial q_N \partial q_s}\dot{q}_s\right)\right]\dot{\mathbf{q}}+ \\ \left[\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{R}_{C_0}}{\partial q_1 \partial q_s}\dot{q}_s\right)\cdot\sum_k m_k\frac{\partial \mathbf{r}_k}{\partial q_i} \quad \cdots \quad \left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{R}_{C_0}}{\partial q_N \partial q_s}\dot{q}_s\right)\cdot\sum_k m_k\frac{\partial \mathbf{r}_k}{\partial q_i}\right]\dot{\mathbf{q}} \tag{3.20}$$

which describes *format-III*. Note that both $\mathbf{R}_{C_0}$ and $\mathbf{r}_k$ have to be differentiated in the inertial frame.

Next, to obtain the system dynamics model, the original terms in the system kinetic energy as obtained in Eq. (3.10), are substituted into the corresponding format. Then, following the structure of the dynamics model presented in Eq. (3.14), appropriate terms are collected to yield the different elements of the model.

3.2.2.2 Mass Matrix

To obtain the mass matrix **H**, according to Eq. (3.14), the acceleration terms in each of the three formats have to collected. Therefore, H_{ij} is computed by

- ❒ Substituting each term of the system kinetic energy, as expressed in Eq. (3.10), into an appropriate format;
- ❒ Finding the coefficients of $\ddot{\mathbf{q}}$ in the corresponding format;
- ❒ Adding the results, obtained from formats I, II, and III, for each term.
- ❒ Adding the results, obtained for all of the terms.

Leaving aside the details, this procedure eventually yields

$$\begin{aligned} H_{ij} = {} & M\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\frac{\partial \mathbf{R}_{C_0}}{\partial q_j} + \frac{{}^{0}\partial\,\boldsymbol{\omega}_0}{\partial \dot{q}_i}\cdot \mathbf{I}_0 \cdot \frac{{}^{0}\partial\,\boldsymbol{\omega}_0}{\partial \dot{q}_j} + \\ & \sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_i}\cdot\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_j} + \frac{{}^{k}\partial\,\boldsymbol{\omega}_k^{(m)}}{\partial \dot{q}_i}\cdot \mathbf{I}_k^{(m)}\cdot\frac{{}^{k}\partial\,\boldsymbol{\omega}_k^{(m)}}{\partial \dot{q}_j}\right) + \\ & \left(\sum_{m=1}^{n}\sum_{k=1}^{N_m} m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_i}\right)\cdot\frac{\partial \mathbf{R}_{C_0}}{\partial q_j} + \left(\sum_{m=1}^{n}\sum_{k=1}^{N_m} m_k^{(m)}\frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_j}\right)\cdot\frac{\partial \mathbf{R}_{C_0}}{\partial q_i} \end{aligned} \tag{3.21}$$

where $\mathbf{r}_{C_k}^{(m)}$ can be substituted from Eq. (2.20), and $\boldsymbol{\omega}_k^{(m)}$ from Eq. (2.12). Note that consistent with Kane and Levinson (1985) a left superscript on partial derivatives refers to the frame in which the differentiation has to be taken, where for the inertial frame it is left as blank. This is followed in the formulations which are developed next.

3.2.2.3 Vector of Nonlinear Terms

The vector of nonlinear velocity terms in Eq. (3.14), can be computed by dropping the acceleration terms, in each of the obtained formats. So, C_i is computed following the same procedure as described for computation of H_{ij}, by considering the coefficients of $\dot{\mathbf{q}}$ and any other term (except those which correspond to $\ddot{\mathbf{q}}$) in each format. Following such a procedure, it can be obtained that

$$\mathbf{C}(\boldsymbol{\delta}_0, \dot{\boldsymbol{\delta}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{C}_1(\boldsymbol{\delta}_0, \dot{\boldsymbol{\delta}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\,\dot{\mathbf{q}} + \mathbf{C}_2(\boldsymbol{\delta}_0, \dot{\boldsymbol{\delta}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \tag{3.22a}$$

where $\dot{\mathbf{q}} = (\dot{\mathbf{R}}_{C_0}^T, \dot{\boldsymbol{\delta}}_0^T, \dot{\boldsymbol{\theta}}^T)^T$, and

$$\mathrm{C}_{1ij} = \frac{{}^0\partial \boldsymbol{\omega}_0}{\partial \dot{q}_i} \cdot \mathbf{I}_0 \cdot \frac{{}^0\partial \boldsymbol{\omega}_0}{\partial q_j} + \boldsymbol{\omega}_0 \cdot \mathbf{I}_0 \cdot \frac{{}^0\partial^2 \boldsymbol{\omega}_0}{\partial \dot{q}_i \partial q_j} + \frac{\partial \mathbf{R}_{C_0}}{\partial q_i} \cdot \sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)} \sum_{s=1}^{N} \frac{\partial^2 \mathbf{r}_{C_k}^{(m)}}{\partial q_s \partial q_j} \dot{q}_s \right) +$$
$$\sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)} \frac{\partial \mathbf{r}_{C_k}^{(m)}}{\partial q_i} \cdot \left(\sum_{s=1}^{N} \frac{\partial^2 \mathbf{r}_{C_k}^{(m)}}{\partial q_s \partial q_j} \dot{q}_s \right) + \frac{{}^k\partial \boldsymbol{\omega}_k^{(m)}}{\partial \dot{q}_i} \cdot \mathbf{I}_k^{(m)} \cdot \frac{{}^k\partial \boldsymbol{\omega}_k^{(m)}}{\partial q_j} + \boldsymbol{\omega}_k^{(m)} \cdot \mathbf{I}_k^{(m)} \cdot \frac{{}^k\partial^2 \boldsymbol{\omega}_k^{(m)}}{\partial \dot{q}_i \partial q_j} \right) \tag{3.22b}$$

$$\mathrm{C}_{2i} = -\left(\boldsymbol{\omega}_0 \cdot \mathbf{I}_0 \cdot \frac{{}^0\partial \boldsymbol{\omega}_0}{\partial q_i} + \sum_{m=1}^{n}\sum_{k=1}^{N_m} \boldsymbol{\omega}_k^{(m)} \cdot \mathbf{I}_k^{(m)} \cdot \frac{{}^k\partial \boldsymbol{\omega}_k^{(m)}}{\partial q_i} \right) \tag{3.22c}$$

Note that using the relationship between the angular velocity (${}^0\boldsymbol{\omega}_0$) and Euler rates ($\dot{\boldsymbol{\delta}}_0$), given by Eq. (3.7) for the spacecraft, vector $\mathbf{C}_2$ can be combined with the first term of Eq. (3.22a). Then, the vector of nonlinear velocity terms can be written as

$$\mathbf{C}(\boldsymbol{\delta}_0, \dot{\boldsymbol{\delta}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \breve{\mathbf{C}}(\boldsymbol{\delta}_0, \dot{\boldsymbol{\delta}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\,\dot{\mathbf{q}} \tag{3.23}$$

This is a representation of nonlinear velocity terms which is preferred in the development of adaptive control algorithms.

3.2.2.4 Vector of Generalized Forces

As described in Eq. (3.15), if all external forces except the ones applied on the spacecraft are zero, the vector of generalized forces $\mathbf{Q}$ is written as

$$\mathbf{Q} = \mathbf{J}_Q \begin{Bmatrix} {}^0\mathbf{f}_s \\ {}^0\mathbf{n}_s \\ \boldsymbol{\tau}_{K\times 1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0}_{6\times 1} \\ \boldsymbol{\tau}_{K\times 1} \end{Bmatrix} + \mathbf{J}_0{}^T \begin{Bmatrix} {}^0\mathbf{f}_s \\ {}^0\mathbf{n}_s \end{Bmatrix} \tag{3.24}$$

Assuming that ${}^0\mathbf{f}_s$ and ${}^0\mathbf{n}_s$ are applied at the spacecraft center of mass, $\mathbf{J}_0$ is defined as

$$\begin{Bmatrix} {}^{0}\dot{\mathbf{R}}_{C_0} \\ {}^{0}\boldsymbol{\omega}_0 \end{Bmatrix} = \mathbf{J}_0\,\dot{\mathbf{q}} \tag{3.25a}$$

Then, similar to the Jacobian matrix given in Eq. (2.28), $\mathbf{J}_0$ can be obtained as

$$\mathbf{J}_0 = \begin{bmatrix} \mathbf{T}_0^{\,T} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{3\times3} & \mathbf{S}_0 & \mathbf{0}_{3\times K} \end{bmatrix}_{6\times N} \tag{3.25b}$$

Therefore, $\mathbf{J}_Q$ is obtained as

$$\mathbf{J}_Q = \begin{bmatrix} \mathbf{T}_0 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{3\times3} & \mathbf{S}_0^{\,T} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{K\times3} & \mathbf{0}_{K\times3} & \mathbf{1}_{K\times K} \end{bmatrix}_{N\times N} \tag{3.26}$$

which can be substituted into Eq. (3.24) to obtain $\mathbf{Q}$. This completes the derivation of the dynamics model for a multiple arm SFFR with rigid links. Note that the computation of the obtained dynamics equations, can be done either by *numerical* or *symbolical* programming tools. Symbolical derivation, i.e. obtaining the system response using analytical expressions for the dynamics, has been followed in this research work, and will be discussed in Section 3.4.

3.3 Supplementary Issues

In this section, in view of future utilization of the dynamics model in the development of control algorithms, some supplementary issues are discussed. *Quasi-coordinate formulation* of the system dynamics, and the outline of a formulation employing *Euler parameters* for orientation representation are briefly presented. Finally, some unique dynamics characteristics pertaining to space robotic systems are discussed.

3.3.1 Quasi-Coordinate Formulation

3.3.1.1 Problem Statement

The form of equations in (3.1) which results in the dynamics model of Eq. (3.14), is useful in designing an Euler angle based control algorithm, as will be discussed in more details in

Section 4.3. In this case, the vector of generalized coordinates was chosen as $\mathbf{q}=(\mathbf{R}_{C_0}^T, \boldsymbol{\delta}_0^T, \boldsymbol{\theta}^T)^T$. For control reasons, it is also beneficial to obtain the equations of motion using as the vector of generalized speeds $\nu=(\dot{\mathbf{R}}_{C_0}^T, {}^0\boldsymbol{\omega}_0^T, \dot{\boldsymbol{\theta}}^T)^T$, where ${}^0\boldsymbol{\omega}_0$ is the angular velocity of the spacecraft expressed in its own body frame. As it is seen, ν is not equal to $\dot{\mathbf{q}}$ anymore, and the equations of motion have to be modified for this set of variables, resulting in a *quasi-coordinate formulation*. This may be of interest in obtaining a dynamics model for model-based control algorithms, developed based on angular velocity of the spacecraft rather than corresponding Euler angles and rates.

3.3.1.2 Equations of Motion in terms of Quasi-Coordinates

The vector of generalized coordinates, $\mathbf{q}=(\mathbf{R}_{C_0}^T, \boldsymbol{\delta}_0^T, \boldsymbol{\theta}^T)^T$, can be arranged as

$$\mathbf{q}=(\mathbf{q}^{(0)^T}, \mathbf{q}^{(1)^T}, \cdots, \mathbf{q}^{(n)^T})^T \tag{3.27a}$$

where

$$\mathbf{q}^{(0)}=(\mathbf{R}_{C_0}^T, \boldsymbol{\delta}_0^T)^T \tag{3.27b}$$

$$\mathbf{q}^{(m)}=\boldsymbol{\theta}^{(m)}=(\theta_1^{(m)}, \theta_2^{(m)}, \cdots, \theta_{N_m}^{(m)})^T \tag{3.27c}$$

and n is the number of manipulators or appendages to the spacecraft. Then, the system kinetic energy is differentiated with respect to $\dot{\boldsymbol{\delta}}_0$ to yield

$$\frac{\partial T}{\partial \dot{q}_i^{(0)}}=\sum_{k=1}^{3}\frac{\partial T}{\partial\, {}^0\omega_{0_k}}\frac{\partial\, {}^0\omega_{0_k}}{\partial \dot{q}_i^{(0)}} \qquad i=4,5,6 \tag{3.28a}$$

where ${}^0\boldsymbol{\omega}_0 \equiv ({}^0\omega_{0_1}, {}^0\omega_{0_2}, {}^0\omega_{0_3})^T$. Based on Eq. (3.7), this results in

$$\frac{\partial T}{\partial \dot{\boldsymbol{\delta}}_0}=\mathbf{S}_0^T\frac{\partial T}{\partial\, {}^0\boldsymbol{\omega}_0} \tag{3.28b}$$

Therefore, the second three equations of the dynamics model which correspond to the spacecraft orientation, can be obtained as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\, {}^0\boldsymbol{\omega}_0}\right)+\left[{}^0\boldsymbol{\omega}_0\right]^{\times}\frac{\partial T}{\partial\, {}^0\boldsymbol{\omega}_0}-\mathbf{S}_0^{-T}\frac{\partial T}{\partial\, \boldsymbol{\delta}_0}=\mathbf{S}_0^{-T}\mathbf{Q}_\delta^{(0)} \tag{3.29}$$

where $\mathbf{Q}_\delta^{(0)}$ is a 3×1 vector which contains the second three elements of the vector of generalized forces in the previously obtained dynamics model, i.e. corresponding to the spacecraft orientation. Note that from Eqs. (3.24) and (3.26) it is obtained that

$$\mathbf{Q}_\delta^{(0)} = \mathbf{S}_0^T \, {}^0\mathbf{n}_s \tag{3.30a}$$

or

$$\mathbf{S}_0^{-T} \mathbf{Q}_\delta^{(0)} = {}^0\mathbf{n}_s \tag{3.30b}$$

Eq. (3.30b) can be substituted into Eq. (3.29) to yield

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\, {}^0\boldsymbol{\omega}_0}\right) + \left[{}^0\boldsymbol{\omega}_0\right]^\times \frac{\partial T}{\partial\, {}^0\boldsymbol{\omega}_0} - \mathbf{S}_0^{-T}\frac{\partial T}{\partial \boldsymbol{\delta}_0} = {}^0\mathbf{n}_s \tag{3.31}$$

which leads to the *quasi-coordinate formulation* for the dynamics of multiple arm SFFR. This is obtained if the second three equations of the dynamics model described in Eq. (3.14) are substituted by Eq. (3.31). As mentioned before, the result is useful for model-based control algorithms which are developed based on angular velocity of the spacecraft rather than corresponding Euler angles. The main purpose of developing such algorithms is overcoming the non-physical singularities, due to an Euler angle representation of attitude, that correspond to a singular $\mathbf{S}_0$. Therefore, the new model is appropriate, if the system kinetic energy is expressed independently of the spacecraft Euler angles, i.e. $\partial T / \partial \boldsymbol{\delta}_0 = \mathbf{0}$, and Eq. (3.31) can be simplified to

$$\frac{d}{dt}\left(\frac{\partial T}{\partial\, {}^0\boldsymbol{\omega}_0}\right) + \left[{}^0\boldsymbol{\omega}_0\right]^\times \frac{\partial T}{\partial\, {}^0\boldsymbol{\omega}_0} = {}^0\mathbf{n}_s \tag{3.32}$$

which has the form of the Euler equation for a single rigid body.

A more reasonable approach to obtain a suitable dynamics model for such control algorithms, is formulating the system dynamics on the basis of choosing *Euler parameters* for orientation representation, which is discussed next.

3.3.2 Using Euler Parameters as Orientational Coordinates

Choosing *Euler parameters* for orientation representation introduces algebraic constraints to the system dynamics. This is due to the fact that these four parameters are not independent, and obey an algebraic constraint. An independent system of equations of motion can be obtained using the *Natural Orthogonal Complement Method*, which is briefly described next.

3.3.2.1 Basic Definitions

Using Euler parameters to describe the spacecraft rotation results in the following vector of generalized coordinates

$$\mathbf{q} = (\mathbf{R}_{C_0}^T, \boldsymbol{\kappa}^T, \boldsymbol{\theta}^T)^T \tag{3.33a}$$

where $\boldsymbol{\kappa}$ is the vector of Euler parameters describing the spacecraft attitude, and is defined as

$$\boldsymbol{\kappa} = \left(\boldsymbol{\varepsilon}^T, \eta\right)^T \tag{3.33b}$$

where $\boldsymbol{\varepsilon}$ and η are defined as

$$\boldsymbol{\varepsilon} = \mathbf{k}\sin(\frac{\theta_0}{2}) \quad \& \quad \eta = \cos(\frac{\theta_0}{2}) \tag{3.33c}$$

where $\mathbf{k} = \mathbf{T}_0\,\mathbf{k}$ defines a 3×1 unit vector along the spacecraft axis of rotation, and θ_0 describes a simple rotation about this axis, Hughes (1986). It can be seen that the four components of $\boldsymbol{\kappa}$ are not independent, and obey the following constraint

$$\boldsymbol{\kappa}\cdot\boldsymbol{\kappa} = 1 \tag{3.34}$$

The vector of generalized speeds is selected as $\boldsymbol{\nu} = (\dot{\mathbf{R}}_{C_0}^T, {}^0\boldsymbol{\omega}_0^T, \dot{\boldsymbol{\theta}}^T)^T$. It can be shown that (see Angeles (1988))

$$\boldsymbol{\omega}_0 = \mathbf{E}\,\dot{\boldsymbol{\kappa}} \tag{3.35a}$$

and

$$\dot{\boldsymbol{\kappa}} = \frac{1}{4}\mathbf{E}^T\,\boldsymbol{\omega}_0 \tag{3.35b}$$

where

$$\mathbf{E} = 2\begin{bmatrix}\breve{\mathbf{E}} & -\boldsymbol{\varepsilon}\end{bmatrix}_{3\times4} \tag{3.35c}$$

$$\breve{\mathbf{E}} = \eta\,\mathbf{1} + [\boldsymbol{\varepsilon}]^{\times} \tag{3.35d}$$

$\mathbf{1}$ is a 3×3 identity matrix. The spacecraft rotation matrix with respect to the inertial frame, $\mathbf{T}_0$, can be written in terms of Euler parameters as

$$\mathbf{T}_0 = \left(2\eta^2 - 1\right)\mathbf{1} + 2\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T + 2\eta\,[\boldsymbol{\varepsilon}]^{\times} \tag{3.36}$$

Based on Eqs. (3.35), it can be written that

$$\dot{\mathbf{q}} = \boldsymbol{\Phi}\boldsymbol{\nu} \tag{3.37a}$$

where

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{4\times3} & \frac{1}{4}\mathbf{E}^T & \mathbf{0}_{4\times K} \\ \mathbf{0}_{K\times3} & \mathbf{0}_{K\times3} & \mathbf{1}_{K\times K} \end{bmatrix}_{(N+1)\times N} \tag{3.37b}$$

which is used in reducing the dynamics equations, as will be discussed later. Conversely

$$\boldsymbol{\nu} = \boldsymbol{\Psi}\dot{\mathbf{q}} \tag{3.38a}$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{0}_{3\times4} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{3\times3} & \mathbf{E}_{3\times4} & \mathbf{0}_{3\times K} \\ \mathbf{0}_{K\times3} & \mathbf{0}_{K\times4} & \mathbf{1}_{K\times K} \end{bmatrix}_{N\times(N+1)} \tag{3.38b}$$

So

$$\boldsymbol{\Phi}\boldsymbol{\Psi} = \mathbf{1}_{(N+1)\times(N+1)} \quad \& \quad \boldsymbol{\Psi}\boldsymbol{\Phi} = \mathbf{1}_{N\times N} \tag{3.39}$$

The constraint defined by Eq. (3.34) can be differentiated to yield

$$\boldsymbol{\kappa}^T\dot{\boldsymbol{\kappa}} = 0 \tag{3.40a}$$

or

$$\mathbf{a}^T\,\dot{\mathbf{q}} = 0 \tag{3.40b}$$

where $\mathbf{a} = \left(\mathbf{0}_{3\times1}{}^T,\ \boldsymbol{\kappa}^T,\ \mathbf{0}_{K\times1}{}^T\right)^T$ is an (N+1)×1 vector. Next, the general Lagrangian formulation is modified to yield the system dynamics under the described constraint.

3.3.2.2 Constrained Equations of Motion

The general Lagrangian formulation, Eq. (3.1), for the described constrained system can be modified as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right)-\left(\frac{\partial T}{\partial \mathbf{q}}\right)=\mathbf{Q}+\mathbf{Q}_s \tag{3.41a}$$

where $\mathbf{Q}$ is the vector of applied forces/torques, and $\mathbf{Q}_s$ is recognized as the vector of constraint forces/torques which can be written as

$$\mathbf{Q}_s = \Lambda\,\mathbf{a} \tag{3.41b}$$

Λ is a scalar, the so-called *Lagrange multiplier*, and $\mathbf{a}$ describes the single constraint as defined in Eq. (3.40). It should be mentioned that for a system with more than one constraints, $\mathbf{Q}_s$ can be obtained as

$$\mathbf{Q}_s = \sum_{k=1}^{i_s} \Lambda_k \mathbf{a}_k \tag{3.42}$$

where i_s is the number of constraints, Meirovitch (1970).

Eq. (3.41) describes the system dynamics in terms of a set of N+1 constrained coordinates. To obtain an independent system of N equations, this equation has to be modified, which is discussed next.

3.3.2.3 Independent System of Equations

Substituting Eq. (3.37) into Eq. (3.40), yields

$$\mathbf{a}^T\mathbf{\Phi} = \left(\mathbf{\Phi}^T\mathbf{a}\right)^T = 0 \tag{3.43}$$

This means $\mathbf{\Phi}^T$ is an orthogonal complement of $\mathbf{a}$, and leads to the concept of *Natural Orthogonal Complement Method* in obtaining an independent system of equations from a constrained system (Saha and Angeles (1991), Cyril et. al. (1991)). Clearly, multiplying Eq. (3.41) by $\mathbf{\Phi}^T$ makes the vector of constraint forces vanish, and yields

$$\Phi^T \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right) - \Phi^T\left(\frac{\partial T}{\partial \mathbf{q}}\right) = \Phi^T \mathbf{Q} \tag{3.44}$$

which represents a system of N independent equations.

To write Eq. (3.44) in terms of generalized speeds ν, $\partial T / \partial \dot{\mathbf{q}}$ can be substituted by

$$\frac{\partial T}{\partial \dot{\mathbf{q}}} = \mathbf{P}\frac{\partial T}{\partial \nu} \tag{3.45a}$$

where

$$\mathbf{P} = \begin{bmatrix} \frac{\partial \nu_1}{\partial \dot{q}_1} & \cdots & \frac{\partial \nu_N}{\partial \dot{q}_1} \\ \vdots & & \vdots \\ \frac{\partial \nu_1}{\partial \dot{q}_{N+1}} & \cdots & \frac{\partial \nu_N}{\partial \dot{q}_{N+1}} \end{bmatrix}_{(N+1)\times N} \tag{3.45b}$$

Considering Eq. (3.38), it can be seen that

$$\mathbf{P} = \Psi^T \tag{3.46}$$

Therefore

$$\frac{\partial T}{\partial \dot{\mathbf{q}}} = \Psi^T \frac{\partial T}{\partial \nu} \tag{3.47}$$

which can be differentiated with respect to time, to yield

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}}\right) = \dot{\Psi}^T \frac{\partial T}{\partial \nu} + \Psi^T \frac{d}{dt}\left(\frac{\partial T}{\partial \nu}\right) \tag{3.48}$$

Substituting Eq. (3.48) into Eq.(3.44), and using Eq. (3.39), results in

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \nu}\right) + \left(\dot{\Psi}\Phi\right)^T \frac{\partial T}{\partial \nu} - \Phi^T\left(\frac{\partial T}{\partial \mathbf{q}}\right) = \Phi^T \mathbf{Q} \tag{3.49}$$

which is a set of N independent equations, and represents the system dynamics in terms generalized speeds selected as $\nu = (\dot{\mathbf{R}}_{C_0}^T, {}^0\omega_0^T, \dot{\theta}^T)^T$, and the generalized coordinates as defined in Eq. (3.33).

Next, in view of future utilization of the dynamics model in development of control algorithms, some specific characteristics of space robotic systems compared to fixed-base manipulators are pointed out.

3.3.3 Dynamics Characteristics of SFFR

In space robotic systems, unlike fixed-base manipulators, any motion of a single link creates a reactional motion of the whole system. In free-floating mode, where no external force is applied on the system, the motion is *dynamically constrained*, i.e. total linear and angular momentum of the system is conserved. Also, the Jacobian matrix as obtained in Eq. (2.14) becomes mass dependent. In other words, the inertial linear velocity of an arbitrary point P, and the angular velocity of the corresponding body, is affected by mass distribution over the entire system. Surprisingly, this coupling between arms and the free base also affects the dynamics of the relative motion of the end-effector with respect to the base. This is due to the fact that joint angles and rates are dynamically coupled, even though the relative motion can be expressed in terms of a fixed-base type Jacobian.

To observe specific characteristics of space robotic systems vigorously, elements of the dynamics model for a fixed-base manipulator are next compared to those of a space robotic system. As shown in Asada and Slotine (1986), for a fixed-base serial manipulator, the mass matrix **H** and the vector of nonlinear velocity terms **C** can be obtained as

$$\mathbf{H} = \sum_{i=1}^{N}\left(m_i \mathbf{J}_L^{(i)^T} \mathbf{J}_L^{(i)} + \mathbf{J}_A^{(i)^T}\, {}^0\mathbf{I}_i^{CM_i}\, \mathbf{J}_A^{(i)}\right) \tag{3.50a}$$

$$C_i = \sum_{j=1}^{N}\sum_{k=1}^{N} m_{ijk}\, \dot{q}_k\, \dot{q}_j \tag{3.50b}$$

where

$$\mathbf{J}_L^{(i)} = \left[\left(\left[{}^0\mathbf{z}_1\right]^{\times}\, {}^0\mathbf{P}_{CM_i}^{1}\right) \;\cdots\; \left(\left[{}^0\mathbf{z}_i\right]^{\times}\, {}^0\mathbf{P}_{CM_i}^{i}\right) \;\; \mathbf{0}_{3\times 1} \;\cdots\; \mathbf{0}_{3\times 1}\right]_{3\times N} \tag{3.51a}$$

$$\mathbf{J}_A^{(i)} = \left[{}^0\mathbf{z}_1 \;\cdots\; {}^0\mathbf{z}_i \;\; \mathbf{0}_{3\times 1} \;\cdots\; \mathbf{0}_{3\times 1}\right]_{3\times N} \tag{3.51b}$$

$$m_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial H_{jk}}{\partial q_i} \tag{3.51c}$$

and

$${}^0\mathbf{P}_{CM_i}^{j} = {}^0\mathbf{T}_j\left({}^j\mathbf{P}_{CM_i}^{j}\right) \tag{3.51d}$$

m_i is the i-th link mass, ${}^0\mathbf{I}_i^{CM_i}$ is its inertia matrix with respect to the center of mass expressed in the fixed frame, ${}^0\mathbf{z}_i$ is a unit vector along the i-th joint axis expressed in the fixed frame, ${}^j\mathbf{P}^j_{CM_i}$ is the position vector of the i-th center of mass with respect to the origin the of j-th frame as seen in that frame, and ${}^0\mathbf{T}_j$ is the rotation matrix between the j-th frame and the fixed one. It can be proven that the obtained H_{ij} and C_i for a fixed-base manipulator, are functions of specific set of mass parameters as

$$H_{ij} = h_{ij}(\tilde{m}_k,\cdots,\tilde{m}_N)\, h'_{ij}(\boldsymbol{\theta}_\bullet) \qquad k = \max(i,j) \tag{3.52a}$$

$$C_i = f_i(\tilde{m}_i,\cdots,\tilde{m}_N)\, f'_i(\boldsymbol{\theta}_\bullet,\dot{\boldsymbol{\theta}}_\bullet) \tag{3.52b}$$

where h_{ij}, h'_{ij}, f_i, and f'_i are functions of the given arguments, $\tilde{m}_i$ denotes the i-th link mass properties (both mass and moment of inertia), and $\boldsymbol{\theta}_\bullet$ is a subset of joint angles vector ($\boldsymbol{\theta}$). As it is seen mass properties have a backward propagation effect on the dynamics model. In other words, mass properties of link "i" do not appear in the $\mathbf{H}$ elements which correspond to posterior joint variables, i.e. i+1,..., N. For instance, mass properties of the first link only appear in H_{11} and C_1. On the contrary, for space manipulators in the *free-floating mode*, this is no longer true, and every element of the dynamics model is affected by mass properties of all links. This can be justified by considering the mass matrix $\mathbf{H}$, and the vector of nonlinear velocity terms $\mathbf{C}$, when obtained based on barycentric vector kinematics. To complete this discussion, the mass matrix $\mathbf{H}$ for a space manipulator, is now presented in terms of barycentric vectors.

Based on Eqs. (3.5) for the system kinetic energy expressed in terms of barycentric vectors, following the same procedure explained in Section 3.2.2.2, it can be obtained

$$\begin{aligned} H_{ij} = {} & M\frac{\partial \mathbf{R}_{CM}}{\partial q_i}\cdot\frac{\partial \mathbf{R}_{CM}}{\partial q_j} + m_0\frac{\partial \boldsymbol{\rho}_{C_0}}{\partial q_i}\cdot\frac{\partial \boldsymbol{\rho}_{C_0}}{\partial q_j} + \frac{{}^0\partial\,\boldsymbol{\omega}_0}{\partial \dot{q}_i}\cdot\mathbf{I}_0\cdot\frac{{}^0\partial\,\boldsymbol{\omega}_0}{\partial \dot{q}_j} + \\ & \sum_{m=1}^{n}\sum_{k=1}^{N_m}\left(m_k^{(m)}\frac{\partial \boldsymbol{\rho}^{(m)}_{C_k}}{\partial q_i}\cdot\frac{\partial \boldsymbol{\rho}^{(m)}_{C_k}}{\partial q_j} + \frac{{}^k\partial\,\boldsymbol{\omega}_k^{(m)}}{\partial \dot{q}_i}\cdot\mathbf{I}_k^{(m)}\cdot\frac{{}^k\partial\,\boldsymbol{\omega}_k^{(m)}}{\partial \dot{q}_j}\right) \end{aligned} \tag{3.53}$$

where ρ_{C_0} and $\rho_{C_k}^{(m)}$ can be substituted from Eq. (2.5), and angular velocities from Eq. (2.12), premultiplying each term by appropriate transformation matrix.

Note that ρ_{C_0} and $\rho_{C_k}^{(m)}$ are written in terms of barycentric vectors, which according to the definition consist of the vectors on every single link of the system. So, ρ_{C_0} and $\rho_{C_k}^{(m)}$ are functions of all joint variables and spacecraft Euler angles, so that $\partial\rho_{C_0}/\partial q_i$ or $\partial\rho_{C_k}^{(m)}/\partial q_i$ results in a non-zero value (for i>3). This means every element of the mass matrix **H** for the space manipulator itself, decoupled from the first six equations which describe the system's translation and spacecraft rotation, is affected by the mass properties of all links. The same conclusion can be made by considering the vector of nonlinear velocity terms **C**. As a consequence of this complexity, namely dependency of every element of the dynamics model and Jacobian matrix on mass properties of all links, any error in the estimation of mass parameters has a more drastic effect on the performance of model-based control algorithms in space.

In *free-flying mode*, where external forces (thrusters, etc.) are applied on the system, the motion is no longer dynamically constrained. Therefore, the end-effector can be moved either by joints motion or the spacecraft motion, resulting in a redundant system. However, manipulators dynamics are coupled through the connected spacecraft, so they are affected by the mass properties of all links. This makes coordinated control of the spacecraft and the attached manipulators an interesting problem.

3.4 Generation of Symbolic Code for Dynamics

3.4.1 Symbolical vs. Numerical Code Generation

As mentioned before, computation of the obtained dynamics equations can be done either numerically or symbolically. The latter is chosen in this research work, and is described here. However, to compare the two programming approaches, the required steps in the numerical computation of the obtained dynamics, is first reviewed. To this end, preparation of few sample terms, i.e. ${}^k\partial\omega_k^{(m)}/\partial q_l$ and ${}^k\partial\omega_k^{(m)}/\partial\dot{q}_l$, for numerical computer programming is discussed. In a similar way, other terms in H_{ij}, C_i, and $\mathbf{J}_Q$ can be obtained, and programmed in the corresponding environment.

First, preliminary calculations for numerical computer programming of ${}^k\partial\omega_k^{(m)}/\partial q_l$ and ${}^k\partial\omega_k^{(m)}/\partial\dot{q}_l$ is presented. Following the arrangement of Eqs. (3.27) for the vector of generalized coordinates, the angular velocity of the k-th link of the m-th manipulator expressed in its own body-fixed frame, ${}^k\omega_k^{(m)}$, can be obtained by substituting Eq. (3.7) into Eq. (2.12) and expressing the result in the corresponding frame. This yields

$$\begin{aligned}{}^k\omega_k^{(m)} &= {}^{k-1}\mathbf{T}_k^{(m)^T}\ {}^{k-2}\mathbf{T}_{k-1}^{(m)^T}\cdots{}^{0}\mathbf{T}_1^{(m)^T}\mathbf{S}_0\,\dot{\boldsymbol{\delta}}_0 + \\ &\sum_{s=l}^{k-l}\left({}^{k-1}\mathbf{T}_k^{(m)^T}\ {}^{k-2}\mathbf{T}_{k-1}^{(m)^T}\cdots{}^{s}\mathbf{T}_{s+1}^{(m)^T}\dot{q}_s^{(m)}\ {}^{s}\mathbf{z}_s^{(m)}\right)+\dot{q}_k^{(m)}\ {}^{k}\mathbf{z}_k^{(m)}\end{aligned} \tag{3.54}$$

where $\mathbf{S}_0$ has been already defined in Eq. (3.7), ${}^{i-1}\mathbf{T}_i^{(m)}$ is a rotation matrix between the i-th body-fixed frame and the previous one, and ${}^i\mathbf{z}_i^{(m)}\equiv(0,0,1)^T$ is a unit vector along the axis of rotation of the i-th joint of the m-th manipulator expressed in its own body-fixed frame. Therefore, it can be obtained

$$\frac{{}^k\partial\omega_k^{(m)}}{\partial q_i^{(p)}}=\begin{cases}\sigma_1 & if\quad p=0\\ 0 & if\quad (p\neq 0\ \&\ p\neq m)\\ \sigma_2 & if\quad (p=m\ \&\ i<k)\\ 0 & if\quad (p=m\ \&\ i>k)\\ \sigma_3 & if\quad (p=m\ \&\ i=k)\end{cases} \tag{3.55}$$

where

$$\sigma_1={}^{k-1}\mathbf{T}_k^{(m)^T}\ {}^{k-2}\mathbf{T}_{k-1}^{(m)^T}\cdots{}^{0}\mathbf{T}_1^{(m)^T}\frac{\partial\mathbf{S}_0}{\partial q_i^{(0)}}\dot{\boldsymbol{\delta}}_0 \tag{3.56a}$$

$$\begin{aligned}\sigma_2 &= {}^{k-1}\mathbf{T}_k^{(m)^T}\cdots{}^{i}\mathbf{T}_{i+1}^{(m)^T}\left(\frac{\partial\ {}^{i-1}\mathbf{T}_i^{(m)^T}}{\partial q_i^{(m)}}\right){}^{i-2}\mathbf{T}_{i-1}^{(m)^T}\cdots{}^{0}\mathbf{T}_1^{(m)^T}\mathbf{S}_0\,\dot{\boldsymbol{\delta}}_0 + \\ &\sum_{s=l}^{i-l}\left({}^{k-1}\mathbf{T}_k^{(m)^T}\cdots{}^{i}\mathbf{T}_{i+1}^{(m)^T}\left(\frac{\partial\ {}^{i-1}\mathbf{T}_i^{(m)^T}}{\partial q_i^{(m)}}\right){}^{i-2}\mathbf{T}_{i-1}^{(m)^T}\cdots{}^{s}\mathbf{T}_{s+1}^{(m)^T}\dot{q}_s^{(m)}\ {}^{s}\mathbf{z}_s^{(m)}\right)\end{aligned} \tag{3.56b}$$

$$\sigma_3 = \frac{\partial\, {}^{k-1}\mathbf{T}_k^{(m)^T}}{\partial q_k^{(m)}}\, {}^{k-2}\mathbf{T}_{k-1}^{(m)^T} \cdots {}^{0}\mathbf{T}_1^{(m)^T} \mathbf{S}_0\, \dot{\delta}_0 + \sum_{s=l}^{k-l}\left(\frac{\partial\, {}^{k-1}\mathbf{T}_k^{(m)^T}}{\partial q_k^{(m)}}\, {}^{k-2}\mathbf{T}_{k-1}^{(m)^T} \cdots {}^{s}\mathbf{T}_{s+1}^{(m)^T} \dot{q}_s^{(m)}\; {}^{s}\mathbf{z}_s^{(m)} \right) \tag{3.56c}$$

Similarly, it can be obtained

$$\frac{{}^k\partial\, \omega_k^{(m)}}{\partial \dot{q}_i^{(p)}} = \begin{cases} \sigma_1' & \text{if } \; p=0 \\ 0 & \text{if } \; (p \neq 0 \;\&\; p \neq m) \\ \sigma_2' & \text{if } \; (p=m \;\&\; i<k) \\ 0 & \text{if } \; (p=m \;\&\; i>k) \\ \sigma_3' & \text{if } \; (p=m \;\&\; i=k) \end{cases} \tag{3.57}$$

where

$$\sigma_1' = {}^{k-1}\mathbf{T}_k^{(m)^T}\; {}^{k-2}\mathbf{T}_{k-1}^{(m)^T} \cdots {}^{0}\mathbf{T}_1^{(m)^T} \mathbf{S}_0 \frac{\partial \dot{\delta}_0}{\partial \dot{q}_i^{(0)}} \tag{3.58a}$$

$$\sigma_2' = {}^{k-1}\mathbf{T}_k^{(m)^T}\; {}^{k-2}\mathbf{T}_{k-1}^{(m)^T} \cdots {}^{i}\mathbf{T}_{i+1}^{(m)^T}\; {}^{i}\mathbf{z}_i^{(m)} \tag{3.58b}$$

$$\sigma_3' = {}^{k}\mathbf{z}_k^{(m)} \tag{3.58c}$$

Note that $\mathbf{S}_0$ is a function of δ_0, and ${}^{i-1}\mathbf{T}_i^{(m)}$ is just a function of $q_i^{(m)}$. Therefore, $\partial\, {}^{i-1}\mathbf{T}_i^{(m)^T} / \partial q_i^{(m)}$, $\partial \mathbf{S}_0 / \partial q_i^{(0)}$, and $\partial \dot{\delta}_0 / \partial \dot{q}_i^{(0)}$ can be calculated analytically, and substituted into Eqs. (3.56) and (3.58). Other terms in H_{ij}, C_i, and $\mathbf{J}_Q$ can also be calculated, in a similar way. The obtained results can then be programmed in a numerical environment, to quantify the system dynamics.

Although numerical derivation seems a cumbersome procedure, it would be the only choice if symbolical programming tools were not available[3]. However, by means of symbolical tools, each term can be analytically calculated in a computer program. For instance, Eq. (3.54) can be directly computerized to represent ${}^k\omega_k^{(m)}$. Then, $\partial\, {}^k\omega_k^{(m)} / \partial q_i$

3- Note that for the numerical development of the dynamic properties of mechanical manipulators, the proposed recursive algorithms can be followed. These algorithms utilize the iterative routines for inverse dynamics, and joint forces and torques measurements, to solve direct dynamics. For further details, one can see a comparison of different methods for developing the dynamics of rigid-body systems presented by Ju and Mansour (1989). Here, the focus is on the computation of the explicit dynamics model obtained based on Lagrange formulation.

and $\partial\, {}^k\boldsymbol{\omega}_k^{(m)} / \partial \dot{q}_i$ will be analytically calculated in a single step, rather than going through different options in Eqs. (3.55) and (3.57). Furthermore, using various mathematical identities and factorization techniques, the result can be simplified to shrink the obtained analytical expressions. Therefore, as mentioned earlier, the symbolical derivation of dynamics model is pursued in this research work, and the developed code is introduced in the next section.

3.4.2 Description of the Code

The derivation of the dynamics equations of motion has been programmed in a symbolic environment (MAPLE), for a multiple manipulator SFFR with rigid elements in a general configuration. The output of the code includes the mass matrix **H**, the vector of nonlinear velocity terms **C**, the Jacobian matrix $\mathbf{J}_Q$ to describe the vector of generalized forces, Jacobian matrix $\mathbf{J}_c$ which describes the task space (employed in control) and its time derivative $\dot{\mathbf{J}}_c$, each one as an analytical function of generalized coordinates/speeds.

The program is initiated by determining the system general configuration, i.e. number of manipulators/appendages, number of links for each one, and degrees-of-freedom for the spacecraft (i.e. three for planar motion, or six for spatial motion). Then, mass properties and geometric parameters for each element of the system have to be specified. These parameters can be substituted by numerical magnitudes or left as parameters. The latter results in long expressions, while the first one yields more concise results particularly when some components of geometric vectors or inertia matrices are zero. In fact, in most studies the dynamics has to be modelled for a specific system and then employed in simulation and control investigations. Usually, for these investigations, the simulation routine has to be run tens of times. Therefore, it is preferable to substitute numerical magnitudes for the system parameters in the dynamics model at the very beginning and make it more concise. The cost is just running the symbolic code, once some desired changes in the system parameters have to be made.

The CM relative position/velocity of each particular body ($\mathbf{r}_{C_i}^{(m)}$, $\dot{\mathbf{r}}_{C_i}^{(m)}$) are computed based on Eqs. (2.20), (2.23). The angular velocity of each particular body expressed in its own body-fixed frame, ${}^{k}\boldsymbol{\omega}_k^{(m)}$, is computed based on Eq. (3.54). Then, the mass matrix **H**, and the vector of nonlinear velocity terms **C**, are computed on the basis of Eqs. (3.21) and (3.22). To obtain concise results, first each vector in these equations (e.g. $\partial\mathbf{r}_{C_i}^{(m)}/\partial q_i$, $\partial\,{}^{k}\boldsymbol{\omega}_k^{(m)}/\partial q_i$ and $\partial\,{}^{k}\boldsymbol{\omega}_k^{(m)}/\partial \dot{q}_i$, etc.) is computed, and only its non-zero components are named and saved as intermediate variables. Then, H_{ij} and C_i are computed and expressed in terms of these intermediate variables, rather than substituting the obtained analytical expression for each one. Jacobian matrices $\mathbf{J}_Q$ and $\mathbf{J}_c$, and the time derivative of the one used in control, $\dot{\mathbf{J}}_c$, are computed similarly.

To simplify the obtained analytical expressions, at each intermediate step, mathematical tools and factorization techniques available in MAPLE, are used. The result of this fairly refined code is a compact analytical dynamics model of the given multiple manipulator SFFR with rigid elements, in terms of generalized coordinates/speeds. Before using this model in simulation and control investigations, it has to be verified as discussed next.

3.4.3 Verification Procedure

The model derivation code, has been run for fixed-base systems which represent limiting cases of space robotic systems, for instance letting the spacecraft mass go to infinity. The output results are verified by comparisons to those calculated by hand. However, since in these limiting cases most of the terms in the dynamics equations vanish, the model has to be also verified in a general case, i.e. for a multiple manipulator space robotic system. This is done by developing another simpler code at a very fundamental level, and comparing the numerical results of the two.

The simpler code, is based on computing the system kinetic energy, using Eq. (3.10), and on its direct substitution into the equations of motion, Eq. (3.1). Obviously, such code yields non-compact equations of motion, compared to those of the code developed and described earlier. However, the simplicity of this code makes it fairly reliable, so that it can be employed as a yardstick for the verification of the developed code which is used in control and simulation. In fact, this was a very helpful approach in finding minor mistakes at various levels, and verifying the developed code at the end.

Table 2.1 shows the difference between typical results obtained from the two codes, for a 14-DOF space robotic system including three manipulators and appendages as described in Section 4.4.3. As it is seen, the difference between obtained vectors of nonlinear velocity terms ($\Delta\mathbf{C}$), and a few sample columns of two mass matrices ($\Delta\mathbf{H}$) are either exactly or approximately (due to truncations) zero. Although these results correspond to a single random set of generalized coordinates/speeds (with non-zero entries), the differences are in the same order of magnitude for several other trials. Therefore, it can be concluded that the developed dynamics modelling code is free of errors, yielding a system of compact equations of motion in terms of system variables. To conclude this chapter, a review of the discussed issues and obtained results is presented next.

Table 3.1: The result of verification procedure.

Row	ΔC	ΔH				
		1-St. column	2-nd Column	3-rd Column	4-th column	14-th column
1	-0.13E-14	0.0	0.0	0.0	0.11E-13	0.0
2	0.18E-14	0.0	0.0	0.0	0.71E-14	0.0
3	-0.47E-14	0.0	0.0	0.0	0.0	0.0
4	-0.27E-14	0.11E-13	0.71E-14	0.0	0.14E-13	0.56E-16
5	-0.27E-14	0.18E-14	0.0	-0.14E-13	0.18E-14	0.28E-16
6	0.18E-14	0.10E-14	-0.18E-14	0.0	-0.36E-14	0.0
7	0.0	0.0	-0.78E-15	0.0	-0.18E-14	0.0
8	-0.44E-15	-0.83E-15	0.36E-14	-0.67E-15	0.36E-14	0.0
9	0.0	-0.67E-15	0.44E-15	-0.22E-15	0.0	0.0
10	-0.13E-14	0.11E-14	0.89E-15	-0.89E-15	0.18E-14	0.0
11	0.17E-15	0.0	-0.18E-14	-0.44E-15	-0.38E-14	0.0
12	0.39E-15	0.0	-0.39E-15	0.11E-15	-0.78E-15	0.0
13	0.28E-16	0.0	0.0	0.0	-0.28E-16	-0.69E-17
14	-0.35E-16	0.0	0.0	0.0	0.56E-16	0.28E-16

3.5 Summary

To obtain the dynamics model of a multiple manipulator SFFR with rigid elements, the general Lagrangian formulation was applied. The system kinetic energy was computed based on the two different kinematics approaches developed in the previous chapter. Comparing the obtained results, the *direct path method* was chosen to develop an explicit dynamics model of the system. Mathematical analyses were implemented for typical terms of the system kinetic energy, and three *formats* were identified and used to differentiate expressions. Next, separate derivations for the mass matrix, vector of nonlinear velocity terms, and generalized forces were presented, and the obtained results were assembled to develop the dynamics model.

In view of future utilization of the dynamics model in development of control algorithms, some supplementary issues were discussed next. The main concern was obtaining an appropriate dynamics model for developing model-based control algorithms which aim at overcoming the non-physical singularities due to Euler angle representation of attitude. To this end, the *Quasi-coordinate formulation* of the system dynamics, also using *Euler parameters* for orientation representation were discussed. The latter introduces algebraic constraints to the system dynamics, and therefore, to obtain independent system of equations of motion, the *Natural Orthogonal Complement Method* was used and briefly described. Next, investigating specific characteristics of space robotic systems, it was shown that any error in the estimation of mass parameters has a more drastic effect on the performance of *model-based control algorithms* in space.

Computation of the obtained dynamics can be done either by *numerical* or *symbolical* programming tools. It was shown that preparation of each term for numerical programming requires cumbersome calculations, while by means of the symbolical tools, each term can be analytically calculated. Also, using various mathematical identities and factorization techniques, the result can be simplified to reduce the obtained analytical expressions.

Therefore, derivation of the dynamics equations has been programmed in a symbolic environment (MAPLE), for a general multiple manipulator space robotic system with rigid elements. The developed code was verified in a general case, by developing another simpler code, and comparing the numerical results of the two.

Chapter 4

Coordinated Motion Control of Multi-Arm Space Free-Flyers

4.1 Introduction

The problem of controlling mechanical manipulators is challenging because of the strong nonlinearities and couplings in the equations of motion. As discussed in Section 3.3.3, in contrast to fixed-base manipulators, in space every element of the mass and Jacobian matrices depend on the mass properties of all the links (free-floating mode), or redundancy is added due to spacecraft degrees-of-freedom (free-flying mode). These characteristics of space manipulators make coordinated control of a spacecraft and its attached manipulators more challenging, compared to fixed-base robotic systems. In this chapter, coordination between a spacecraft and its several end-effectors, is investigated under different control laws during a capture maneuver of moving objects in space.

To ensure smooth operation, and to reduce disturbances on the spacecraft and on the object just before grasping, appropriate trajectories for the spacecraft and its manipulators are planned, Section 4.2. Two model-based control algorithms, based on an *Euler angle* and an *Euler parameter* description of the orientation, and a transpose Jacobian control algorithm are developed in Section 4.3. These algorithms permit control of both the spacecraft and its appendages in their task space. The Euler angle model-based control algorithm, called for brevity MB1, presents the inconvenience of representational singularities, while the Euler parameter model-based control algorithm (MB2) overcomes these non-physical singularities.

Next, the performance of the model-based algorithms is compared by simulation, to that of a transpose Jacobian algorithm. First, the verification procedure of the simulation code is discussed. Then, employing a planar example, the importance of a symmetric vs. a non-symmetric grasp, and the ratio of spacecraft maximum acceleration/ deceleration is investigated by simulation. The performance of the MB and TJ algorithms is discussed during two and three-dimensional maneuvers. Results show that due to the complexity of space robotic systems, a drastic deterioration in the performance of model-based algorithms results in the presence of model uncertainties. In such cases, a simple transpose Jacobian algorithm yields comparable results with reduced computational burden, an issue which is very important in space. A summary of the discussed issues and obtained results, in Section 4.5, will conclude this chapter.

4.2 Trajectory Planning

In this section, appropriate trajectories for the spacecraft and its manipulators are planned to result in capturing moving space objects, assumed to be passive. These trajectories ensure smooth operation, and reduce disturbances on the spacecraft and on the object just before grasping. For the spacecraft motion, in both translation and rotation, parabolic trajectories are planned. The manipulators remain in their home configuration as long as the final position of the object is not in their fixed-base reachable workspace. When the object enters the reachable workspace of an end-effector[4], a quintic trajectory is planned in the task space

4- The planned trajectory for the spacecraft rotation aims to provide a symmetric grasp of the object, by two participating manipulators. Therefore, the object enters the fixed-base workspace of both end-effectors, almost at the same time.

to capture the object. All of these trajectories which are discussed next, take into account the relative target motion, and thruster or actuator saturation limits.

4.2.1 Spacecraft Position and Orientation Trajectories

For the spacecraft motion, in both translation and rotation, parabolic trajectories made of constant acceleration, constant velocity, and constant deceleration segments are planned. Since the object detecting sensors are usually on board, and thruster capabilities can be directly converted to the spacecraft maximum acceleration and deceleration magnitudes in the body frame, the desired trajectories are first planned in the spacecraft frame at initial time. These trajectories are subsequently transformed to the inertial space.

For instance, considering translational motion, ${}^0\mathbf{x}_0(t)=[{}^0x_0, {}^0y_0, {}^0z_0]^T$ denotes the desired trajectory for the spacecraft CM position expressed in the body-fixed frame at initial time. To plan the desired trajectories, a motion final time, t_f, is first selected. During capture, it is desired to have the object stationery in the spacecraft frame. Therefore, the desired spacecraft velocity at final time, ${}^0\mathbf{v}_f$, is chosen as

$$ {}^0\mathbf{v}_f = {}^0\mathbf{v}^0_{obj}(0) + {}^0\mathbf{v}_0(0) \tag{4.1} $$

and the desired final position of the spacecraft CM, ${}^0\mathbf{x}_f$, is given by

$$ {}^0\mathbf{x}_f = {}^0\mathbf{x}^0_{obj}(0) + {}^0\mathbf{v}_f\, t_f + {}^0\mathbf{r} \tag{4.2} $$

where ${}^0\mathbf{v}_0(0)$ is the initial velocity of the spacecraft, ${}^0\mathbf{x}^0_{obj}(0)$ and ${}^0\mathbf{v}^0_{obj}(0)$ are the position and velocity of the object as measured *with respect to* the spacecraft CM at initial time and expressed in the body frame, and ${}^0\mathbf{r}$ defines the relative position of the spacecraft CM and a point of interest on the object at time t_f. The direction of ${}^0\mathbf{r}$ is calculated along the line connecting the spacecraft CM at initial time with the object location at t_f, and its magnitude is such that the manipulators can dexterously reach the object.

Next, parabolic trajectories made of constant acceleration, constant velocity, and constant deceleration segments, are planned to yield a final position equal to ${}^0\mathbf{x}_f$, and a final velocity equal to ${}^0\mathbf{v}_f$, see Figure 4.1. Given the maximum acceleration $\mathbf{a}_1$, and maximum deceleration $\mathbf{a}_2$, using the above expressions, the desired trajectory for the spacecraft CM position, is obtained as

$$
{}^0\mathrm{x}_{0_i}(t) = \begin{cases} 0.5\mathrm{a}_{1_i} t^2 + {}^0\mathrm{v}_{0_i}(0)t & \text{if } \; t < t_{1_i} \\ 0.5\mathrm{a}_{1_i} t_{1_i}{}^2 + {}^0\mathrm{v}_{0_i}(0)t_{1_i} + \left(\mathrm{a}_{1_i} t_{1_i} + {}^0\mathrm{v}_{0_i}(0)\right)\left(t - t_{1_i}\right) & \text{if } \; t_{1_i} < t < t_{2_i} \\ 0.5\mathrm{a}_{1_i} t_{1_i}{}^2 + {}^0\mathrm{v}_{0_i}(0)t_{1_i} + \left(\mathrm{a}_{1_i} t_{1_i} + {}^0\mathrm{v}_{0_i}(0)\right)\left(t - t_{1_i}\right) - 0.5\mathrm{a}_{2_i}\left(t - t_{2_i}\right)^2 & \text{if } \; t_{2_i} < t < t_f \end{cases}
\tag{4.3}
$$

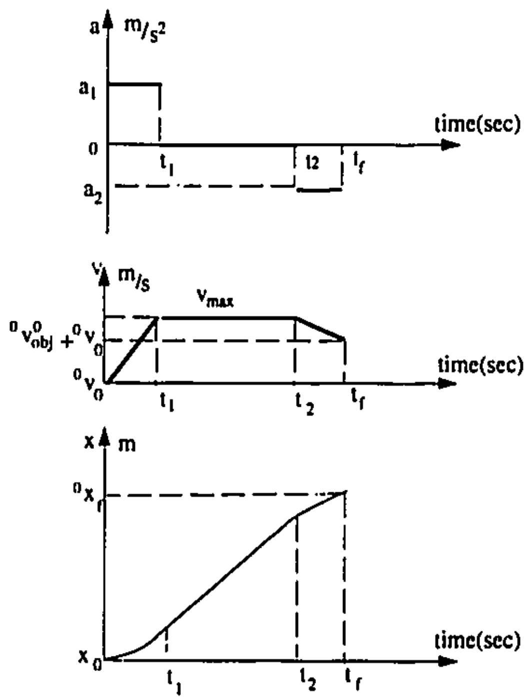


Figure 4.1: Typical profiles of the planned parabolic trajectory.

where subscript "i" describes a relevant component of the corresponding vector. Time t_1 at which the acceleration segment ends, and time t_2 at which the deceleration segment starts, are obtained as

$$
t_{1_i} = \frac{-b_i \pm \sqrt{b_i^2 - 4a_i c_i}}{2a_i} \tag{4.4a}
$$

$$
t_{2_i} = \frac{{}^0\mathrm{v}^0_{obj_i}(0) - \mathrm{a}_{1_i} t_{1_i}}{\mathrm{a}_{2_i}} + t_f \tag{4.4b}
$$

where

$$a_i = 0.5(\mathrm{a}_{1_i} + \frac{\mathrm{a}_{1_i}{}^2}{\mathrm{a}_{2_i}}) \qquad b_i = -\mathrm{a}_{1_i} t_f - \frac{{}^0\mathrm{v}^0_{obj_i}(0)\mathrm{a}_{1_i}}{\mathrm{a}_{2_i}}$$
$$c_i = {}^0\mathrm{x}^0_{obj_i}(0) + {}^0\mathrm{v}^0_{obj_i}(0)\, t_f - {}^0\mathrm{r}_i + 0.5\frac{{}^0\mathrm{v}^0_{obj_i}(0)^2}{\mathrm{a}_{2_i}} \tag{4.4c}$$

Note that the off/on times, t_1 and t_2, are not necessarily equal for all three axes, (i = 1,2,3), corresponding to three components of spacecraft's CM position. Also, in the case of having two positive solutions for t_1, the smaller one is chosen to minimize energy consumption. Estimates for $\mathbf{a}_1$ and $\mathbf{a}_2$ can be obtained using thruster force/torque capabilities and the mass properties of the system.

After computing the desired trajectory in the spacecraft frame at initial time, ${}^0\mathbf{x}_0(t)$, the trajectory in inertial space is computed by

$$\mathbf{x}_0(t) = \mathbf{x}_0(0) + \mathbf{T}_0(0)\ {}^0\mathbf{x}_0(t) \tag{4.5}$$

where $\mathbf{T}_0(0)$ is the rotation matrix between the spacecraft frame (at initial time) and the inertial frame, $\mathbf{x}_0(0)$ is the inertial position of the spacecraft CM at initial time, and $\mathbf{x}_0(t)$ is the inertial trajectory. In practice, the object would be under observation during the chase phase. Should its trajectory change significantly, a new spacecraft chase trajectory would be replanned following the same procedure.

The desired trajectory for the orientation of the spacecraft, is similarly planned. The final orientation is chosen so as to provide an approximately symmetric motion of the manipulators during capture, since this strategy can minimize spacecraft disturbances. To ensure this symmetric motion, the final time for orientational motion is chosen to be smaller than the final time used for the translational motion. Then the desired rotation matrix at final time is assembled such that an axis of symmetry for the spacecraft is aligned with the direction of the object motion. To position the end-effectors, this constraint yields an infinite number of solutions. Therefore another constraint should be added, e.g. keeping the spacecraft roll angle (if the attitude is described by Euler angles) constant during the maneuver. Then, the corresponding parameters for the spacecraft final attitude are extracted from the desired rotation matrix. Having these values, the desired trajectory for the orientation of the spacecraft can be similarly planned.

4.2.2 Manipulator Motion Trajectories

The manipulators remain in their home configuration as long as the final position of the object is not in their fixed-base reachable workspace. During that period, a joint-space controller acting as a brake, is used. When the object enters the reachable workspace of an end-effector, $t = t_r$, a quintic trajectory is planned in the task space for that end-effector, and accordingly a task-space control algorithm is applied. For instance, to plan the desired trajectory for end-effector position, six coefficients have to be determined for each component. First, the end-effector position, linear velocity, and acceleration at starting time ($t = t_r$) are computed based on the current spacecraft position/orientation, and its linear and angular velocity and acceleration. The final values are also computed based on final position and velocity of the object. Then, the six coefficients of the desired quintic trajectory can be computed based on end-effector position, linear velocity, and acceleration at initial and final time, Craig (1989). The result provides continuity of end-effector position, linear velocity, and acceleration, throughout the motion. The desired trajectory for end-effector orientation, can be similarly planned. For some appendages, e.g. the communications antenna, a constant attitude in the inertial frame is commanded throughout the maneuver.

4.3 Control Algorithms Design

Controlling a dynamic system requires definition of the controlled outputs, and design of a control law which can guarantee that these outputs will track desired trajectories asymptotically. For a robotic system, there are various options for the controlled outputs, e.g. joint space variables, Cartesian (task) space variables, and others. The various orientation representations further increase the available options. To control a space free-flying robot (SFFR), different combinations of these options can be chosen. In this research work, the focus is in controlling the Cartesian position and orientation of the spacecraft and the end-effectors of its manipulators.

Coordination between the spacecraft motion and several end-effectors in capturing moving objects in space, is investigated in this section. To this end, two model-based control algorithms, based on an *Euler angle* and on an *Euler parameter* description of the orientation, and a transpose Jacobian control algorithm are developed. Euler angle model-based control algorithm (MB1) presents the inconvenience of representational singularities, i.e. the inversion of the relation between angular velocity and Euler rates, Eq. (3.7), is not possible at some orientations. Such an inversion is required in calculating actuator forces/torques based on the control command which yields the vector of generalized forces $\mathbf{Q}$. In other words, the inversion of Eq. (3.16) is required to find actuator forces/torques, and this is not possible at some orientations. Considering Eq. (3.26), this happens when $\mathbf{S}_0$ becomes singular. So, the orientational error grows as the system approaches these singularities, and if it goes through these points, the control system fails. Therefore, at such points, a different set of Euler angles must be used. It is expected that such singularities will occur whenever a three-parameter description of the orientation is employed. However, a great improvement can occur if a singularity appears at some *attitude error* and not at some *attitude*. An Euler parameter model-based control algorithm that achieves this condition has been presented for the attitude control of a single rigid body, Paielli and Bach (1993). This algorithm is adapted here as part of a coordination scheme to control a multiple arm free-flyer robot, and is presented as the second model-based control algorithm (MB2). Implementation of the model-based control algorithms requires knowledge of the system dynamics, and a considerable computational power. On the other hand, the simpler transpose Jacobian (TJ) controller, as an approximation of MB1, does not require knowledge of the system dynamics and can be employed with less computational burden. In the following, these three algorithms, i.e. two model-based control and the transpose Jacobian control algorithm, are developed and analyzed.

4.3.1 Model-Based Control Design

As discussed in Chapter 3, the dynamics equations for a multiple manipulator SFFR can be obtained as

$$\mathbf{H}(\mathbf{q})\dot{\nu} + \mathbf{C}(\mathbf{q},\nu) = \mathbf{Q}(\mathbf{q}) \tag{4.6}$$

To develop model-based algorithms, on the basis of a feedback linearization approach, a model of system dynamics such as Eq. (4.6) should be employed. Next, assuming that the system geometric and mass properties are known exactly, two model-based control algorithms, based on an Euler angle and on an Euler parameter description of the orientation, are developed.

4.3.1.1 Using Euler Angles (MB1)

Development of the control algorithm is described in three steps. First, the dynamics model is obtained in terms of controlling variables. Then, the control law is introduced, and in the third step, computation of the control command and error behavior are discussed.

Step 1. Assuming that $\mathbf{q} = (\mathbf{R}_{C_0}^T, \boldsymbol{\delta}_0^T, \boldsymbol{\theta}^T)^T$ has been chosen as vector of generalized coordinates, the dynamics model described in Eq. (4.6) can be obtained based on Eq.(3.14). However, the variables to be controlled differ from $\mathbf{q}$, since they include end-effector positions and orientations in Cartesian space. These controlled variables are denoted by $\hat{\mathbf{q}}$ as

$$\hat{\mathbf{q}} = [\mathbf{R}_{C_0}^T, \boldsymbol{\delta}_0^T, \mathbf{x}_E^{(1)T}, \boldsymbol{\delta}_E^{(1)T}, \cdots, \mathbf{x}_E^{(n)T}, \boldsymbol{\delta}_E^{(n)T}]^T \tag{4.7}$$

where $\mathbf{x}_E^{(m)}$ and $\boldsymbol{\delta}_E^{(m)}$ correspond to the m-th end-effector position and orientation.

To develop a model-based algorithm, the dynamics has to be written in terms of $\hat{\mathbf{q}}$. If all manipulators have six DOF, then a space robotic system of n manipulators will have

$6n+6$ DOF, and $\hat{\mathbf{q}}$ will be a $6n+6$ vector. The output speeds $\dot{\hat{\mathbf{q}}}$ are obtained from the generalized speeds $\nu = \dot{\mathbf{q}}$, using a square Jacobian $\mathbf{J}_{c_\delta}$

$$\dot{\hat{\mathbf{q}}} = \mathbf{J}_{c_\delta}(\boldsymbol{\delta}_0, \boldsymbol{\theta})\dot{\mathbf{q}} \tag{4.8}$$

The Jacobian $\mathbf{J}_{c_\delta}$ is not singular, except when a manipulator is at a singular configuration, or at a (non-physical) representation singularity due to the use of Euler angles. The latter can be avoided by switching to a different set of Euler angles. The equations of motion in terms of the output variables, can be obtained as

$$\hat{\mathbf{H}}_\delta \, \ddot{\hat{\mathbf{q}}} + \hat{\mathbf{C}}_\delta = \hat{\mathbf{Q}}_\delta \tag{4.9}$$

where $\hat{\mathbf{H}}_\delta$, $\hat{\mathbf{C}}_\delta$ and $\hat{\mathbf{Q}}_\delta$ are given by

$$\hat{\mathbf{H}}_\delta = \mathbf{J}_{c_\delta}^{-T} \, \mathbf{H} \, \mathbf{J}_{c_\delta}^{-1} \tag{4.10a}$$

$$\hat{\mathbf{C}}_\delta = \mathbf{J}_{c_\delta}^{-T} \, \mathbf{C} - \hat{\mathbf{H}}_\delta \, \dot{\mathbf{J}}_{c_\delta} \, \dot{\mathbf{q}} \tag{4.10b}$$

$$\hat{\mathbf{Q}}_\delta = \mathbf{J}_{c_\delta}^{-T} \, \mathbf{Q} \tag{4.10c}$$

The new inertia matrix, $\hat{\mathbf{H}}_\delta$, is positive definite if $\mathbf{J}_{c_\delta}$ is nonsingular.

__Step 2__. The following model-based control law is used

$$\hat{\mathbf{Q}}_\delta = \hat{\mathbf{H}}_\delta \, \mathbf{u} + \hat{\mathbf{C}}_\delta \tag{4.11}$$

where it is assumed that the system geometric and mass properties are known, and $\mathbf{u} = [\mathbf{u}_{\mathbf{R}_{C0}}^T, \mathbf{u}_\delta^T, \mathbf{u}_x^{(1)T}, \mathbf{u}_\delta^{(1)T}, \cdots, \mathbf{u}_x^{(n)T}, \mathbf{u}_\delta^{(n)T}]$ is an auxiliary control signal which will be determined in Step 3. Substituting Eq. (4.11) into Eq. (4.9), reveals that this control law linearizes and decouples the system equations to a set of second order differential equations

$$\ddot{\hat{\mathbf{q}}} = \mathbf{u} \tag{4.12}$$

__Step 3__. The auxiliary control signal $\mathbf{u}$ can be computed as

$$\mathbf{u} = \mathbf{K}_p \mathbf{e} + \mathbf{K}_d \dot{\mathbf{e}} + \ddot{\hat{\mathbf{q}}}_{des} \tag{4.13}$$

where $\mathbf{K}_p$, and $\mathbf{K}_d$ are chosen as *positive definite* matrices, to result in a guaranteed stable error behavior, and $\mathbf{e}$ is the tracking error defined as

$$\mathbf{e} = \hat{\mathbf{q}}_{des} - \hat{\mathbf{q}} \tag{4.14}$$

Substituting Eq. (4.13) into Eq. (4.12), the control law given by Eq. (4.11) yields

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \, \mathbf{e} = \mathbf{0} \tag{4.15}$$

which guarantees asymptotic convergence of the tracking error $\mathbf{e}$ to zero. Note that $\mathbf{K}_p$, and $\mathbf{K}_d$ can be chosen as *diagonal* matrices, to obtain

$$\ddot{e}_i + K_{d_i} \dot{e}_i + K_{p_i} e_i = 0 \tag{4.16}$$

which decouples the error equations to a set of separated second order differential equations for every single output variable.

The desired trajectory, $\hat{\mathbf{q}}_{des}$, is provided by a trajectory planner, see Section 4.2, while $\hat{\mathbf{q}}$ can be obtained from inertial measurements of the position and orientation of the spacecraft and of the end-effectors[5]. If no such measurements are available, the error $\mathbf{e}$ can be estimated by integrating the equations of motion in real time, but then errors due to model uncertainties will be introduced. A mixed strategy can also be employed, e.g. inertial feedback may be available during a critical or terminal phase of a maneuver.

4.3.1.2 Using Euler parameters (MB2)

Similar to the development of MB1, the MB2 control algorithm is described following the three introduced steps.

__Step 1__. Assuming that $\mathbf{q} = (\mathbf{R}_{C_0}^T, \boldsymbol{\kappa}^T, \boldsymbol{\theta}^T)^T$ defined by Eq. (3.33), has been chosen as vector of generalized coordinates, and $\boldsymbol{\nu} = (\dot{\mathbf{R}}_{C_0}^T, {}^0\boldsymbol{\omega}_0^T, \dot{\boldsymbol{\theta}}^T)^T$ as vector of generalized speeds, the dynamics model described in Eq. (4.6) can be obtained based on Eq. (3.49). Then, it is rewritten in terms of the output speeds $\hat{\boldsymbol{\nu}}$ selected as

$$\hat{\boldsymbol{\nu}} = [\dot{\mathbf{R}}_{C_0}^T, {}^0\boldsymbol{\omega}_0^T, \dot{\mathbf{x}}_E^{(1)T}, {}^1\boldsymbol{\omega}_E^{(1)T}, \cdots, \dot{\mathbf{x}}_E^{(n)T}, {}^n\boldsymbol{\omega}_E^{(n)T}]^T \tag{4.17}$$

5- The end-effector position and orientation can also be computed, based on joint measurements, using manipulator direct kinematics and spacecraft feedback.

where $\dot{\mathbf{x}}_E^{(m)}$ and ${}^m\boldsymbol{\omega}_E^{(m)T}$ are the m-th end-effector linear and angular inertial velocities, expressed in the inertial and m-th end-effector body frame, respectively. If all manipulators have six DOF, then a system of n manipulators will have $6n$+6 DOF, and $\hat{\nu}$ will be a $6n$+6 vector. The output speeds $\hat{\nu}$ are obtained from the generalized speeds ν by a Jacobian $\mathbf{J}_{c_\nu}$

$$\hat{\nu} = \mathbf{J}_{c_\nu}(\boldsymbol{\kappa},\boldsymbol{\theta})\nu \tag{4.18}$$

The equations of motion can be obtained as

$$\hat{\mathbf{H}}_{\hat{\nu}}\,\dot{\hat{\nu}} + \hat{\mathbf{C}}_{\hat{\nu}} = \hat{\mathbf{Q}}_{\hat{\nu}} \tag{4.19}$$

where $\hat{\mathbf{H}}_{\hat{\nu}}$, $\hat{\mathbf{C}}_{\hat{\nu}}$ and $\hat{\mathbf{Q}}_{\hat{\nu}}$ are given by

$$\hat{\mathbf{H}}_{\hat{\nu}} = \mathbf{J}_{c_\nu}^{-T}\,\mathbf{H}\,\mathbf{J}_{c_\nu}^{-1} \tag{4.20a}$$

$$\hat{\mathbf{C}}_{\hat{\nu}} = \mathbf{J}_{c_\nu}^{-T}\,\mathbf{C} - \hat{\mathbf{H}}_{\hat{\nu}}\,\dot{\mathbf{J}}_{c_\nu}\,\dot{\mathbf{q}} \tag{4.20b}$$

$$\hat{\mathbf{Q}}_{\hat{\nu}} = \mathbf{J}_{c_\nu}^{-T}\,\mathbf{Q} \tag{4.20c}$$

<u>Step 2</u>. The following model-based control law, under the assumption of knowledge of the system's properties, is used

$$\hat{\mathbf{Q}}_{\hat{\nu}} = \hat{\mathbf{H}}_{\hat{\nu}}\,\mathbf{u} + \hat{\mathbf{C}}_{\hat{\nu}} \tag{4.21}$$

where **u** is an auxiliary control input which is determined in Step 3. Applying this law to the equations of motion (4.19), results in the following decoupled system

$$\dot{\hat{\nu}} = \mathbf{u} \tag{4.22}$$

Note that Eq. (4.22) is expressed in terms of *linear and angular velocities*, and not in terms of *positions and Euler angles* as is the case in Eq. (4.12).

<u>Step 3</u>. The auxiliary control signal **u** is partitioned as

$$\mathbf{u} = [\mathbf{u}_{\dot{\mathbf{R}}_0}^T, \mathbf{u}_{\omega_0}^T, \mathbf{u}_{\dot{\mathbf{x}}}^{(1)T}, \mathbf{u}_{\omega}^{(1)T}, \ldots, \mathbf{u}_{\dot{\mathbf{x}}}^{(n)T}, \mathbf{u}_{\omega}^{(n)T}]^T \tag{4.23}$$

where the partition follows that of $\hat{\nu}$. The acceleration terms in Eq. (4.22) that correspond to linear motions are controlled similar to Eq. (4.13). For example, $\mathbf{u}_{\dot{\mathbf{R}}_0}$ is given by

$$\mathbf{u}_{\mathbf{R}_0} = \mathbf{K}_{p,\mathbf{R}}\mathbf{e}_{\mathbf{R}} + \mathbf{K}_{d,\mathbf{R}}\dot{\mathbf{e}}_{\mathbf{R}} + \ddot{\mathbf{R}}_{C_0,des} \tag{4.24a}$$

where

$$\mathbf{e}_{\mathbf{R}} = \mathbf{R}_{C_0,des} - \mathbf{R}_{C_0} \tag{4.24b}$$

which according to previous discussion results in

$$\ddot{\mathbf{e}}_{\mathbf{R}} + \mathbf{K}_{d,\mathbf{R}}\dot{\mathbf{e}}_{\mathbf{R}} + \mathbf{K}_{p,\mathbf{R}}\mathbf{e}_{\mathbf{R}} = \mathbf{0} \tag{4.24c}$$

However, to obtain similar asymptotic convergence of attitude error expressed in terms of Euler parameters, the terms that correspond to angular velocities are controlled using

$$\mathbf{u}_{\omega} = \mathbf{T}_e\,\dot{\boldsymbol{\omega}}_{des} + [\boldsymbol{\omega}]^{\times}\mathbf{e}_{\omega} - \mathrm{K}_{d,\omega}\,\mathbf{e}_{\omega} - 2(\mathrm{K}_{p,\omega} - \mathbf{e}_{\omega}^{T}\mathbf{e}_{\omega}/4)\mathbf{e}_{\varepsilon}/e_{\eta} \tag{4.25}$$

$\mathbf{u}_{\omega}$ is expressed in the corresponding body frame. The matrix $\mathbf{T}_e$ relates the error between the desired and current attitude in terms of rotation matrices. In fact, it is a rotation matrix which maps the body frame with desired orientation to the actual body frame, and is defined as

$$\mathbf{T} = \mathbf{T}_e\,\mathbf{T}_{des} \tag{4.26}$$

or

$$\mathbf{T}_e = \mathbf{T}\,\mathbf{T}_{des}^{T} \tag{4.27}$$

The matrix $\mathbf{T}$ is a rotation matrix which corresponds to the current body orientation with respect to the inertial frame, and $\mathbf{T}_{des}$ is the one which corresponds to the desired orientation. The vector $\mathbf{e}_{\omega}$ is the error in angular velocity, expressed in the actual body-fixed frame

$$\mathbf{e}_{\omega} = \boldsymbol{\omega} - \mathbf{T}_e\,\boldsymbol{\omega}_{des} \tag{4.28}$$

where $\boldsymbol{\omega}$ is the current angular velocity of the corresponding body expressed in its own body fixed frame, and $\boldsymbol{\omega}_{des}$ is the desired angular velocity, expressed in the desired orientation frame. So, the term $\mathbf{T}_e\,\boldsymbol{\omega}_{des}$ represents the desired angular velocity resolved in the actual body frame, and the subtraction in Eq. (4.28) is in terms of consistent coordinates. Finally, $\mathbf{e}_{\varepsilon}$ and e_{η}, which correspond to the error in attitude as expressed by Euler parameters, are defined as

$$\mathbf{e}_\varepsilon = \breve{\mathbf{E}}_{des}^T \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_{des}\eta \tag{4.29}$$

$$e_\eta = \boldsymbol{\varepsilon}_{des}^T \boldsymbol{\varepsilon} + \eta_{des}\eta \tag{4.30}$$

where $\breve{\mathbf{E}}$ has been already defined in Eq. (3.35d), repeated here

$$\breve{\mathbf{E}} = \eta\,\mathbf{1} + [\boldsymbol{\varepsilon}]^\times \tag{3.35d}$$

where $\mathbf{1}$ is a 3×3 identity matrix, and $\boldsymbol{\varepsilon}$ and η are the current Euler parameters. Considering Eqs. (4.29, 30), for perfect tracking it can be obtained

$$\left(\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_{des} \quad \& \quad \eta = \eta_{des}\right) \Rightarrow \left(\mathbf{e}_\varepsilon = \mathbf{0} \quad \& \quad e_\eta = 1\right) \tag{4.31}$$

It should be noted that assuming the same axis of rotation (for the desired and actual orientations), the above definitions given for $\mathbf{e}_\varepsilon$ and e_η result in $\|\mathbf{e}_\varepsilon\| = \sin(e_{\theta_0}/2)$ and $e_\eta = \cos(e_{\theta_0}/2)$ where θ_0 describes a simple rotation about axis of rotation[6], and e_{θ_0} is error in θ_0. Therefore, these definitions are geometrically meaningful, rather than $(\boldsymbol{\varepsilon}_{des} - \boldsymbol{\varepsilon})$ and $(\eta_{des} - \eta)$ which do not have any physical interpretation. Also note that due to the form of Eq. (4.25), singularities occur only when e_η is zero, that is when the *attitude error* angle is π rad about the eigen axis, i.e. $e_\eta = \cos(e_{\theta_0}/2) = \cos(\pi/2) = 0$.

Applying the control law given by Eq. (4.25), the attitude error is governed by a homogeneous linear second order differential equation, which guarantees that the error will converge asymptotically to zero

$$\ddot{\mathbf{e}}_\varepsilon + \mathbf{K}_{d,\omega}\,\dot{\mathbf{e}}_\varepsilon + \mathbf{K}_{p,\omega}\,\mathbf{e}_\varepsilon = \mathbf{0} \tag{4.32}$$

In fact, Eq. (4.25) is obtained based on Eq. (4.32), the definitions given for $\mathbf{e}_\varepsilon$ and e_η, and the relationship between angular velocity and Euler parameters as presented in Eq. (3.35), see also Paielli and Bach (1993).

Therefore, considering Eqs. (4.24) and (4.32), it can be concluded that applying the control law given by Eq. (4.21) guarantees asymptotic convergence for the position errors, and attitude error expressed in terms of Euler parameters.

6- As explained below Eq. (3.33).

4.3.2 Transpose Jacobian Algorithm

Considering Eq. (4.11) which describes the model-based algorithm developed as MB1, if high enough gains are used, the simpler transpose Jacobian controller (TJ) can be employed, Craig (1989), as

$$\mathbf{Q} = \mathbf{J}_{C_8}^{T}\left\{\mathbf{K}_p\,\mathbf{e} + \mathbf{K}_d\,\dot{\mathbf{e}}\right\} \tag{4.33}$$

This algorithm is quite simple to use with no significant computational burden, and without requiring a priori knowledge of plant dynamics. However, the Jacobian introduced in Eq. (4.8) which includes system geometric parameters must be used, so that the error is properly resolved. Note that, in fact, this algorithm is an approximation of MB1. Its action can be understood by imagining generalized springs and dampers connected between the bodies under control and the desired trajectories; the stiffer the gains are, the better the tracking should be. If a physical singularity is encountered, the controller given by Eq. (4.33) will result in errors but will not fail computationally.

Next, using Eqs. (4.10-13) and (4.33), the efficiency of the TJ algorithm is compared to the model-based algorithms, in terms of the required computational operations, i.e. multiplication and summations required to follow the algorithm (for an N DOF system). This comparison between the algorithms, in terms of the required computational operations, is depicted in Table 4.1. The model-based algorithm MB1, has been chosen to represent model-based algorithms, although it requires less computational effort compared to MB2. Also, it is assumed that the inverse of the Jacobian matrix and its time derivative, which are required for implementing MB algorithms, are available symbolically. Hence, computations required for inversion of the Jacobian matrix and its time differentiation are not counted. It can be seen that even with these assumptions in favor of the model-based algorithm, implementation of TJ control significantly reduces the amount of required computations, an issue which is very important in space. Stability analysis, based on Lyapunov's theorems, shows that TJ algorithm is asymptotically stable, Section 5.3. As discussed in Section 4.4, the performance of the TJ algorithm is acceptable but deteriorates in tracking fast trajectories. In Chapter 5, further work on this algorithm focuses on reducing this problem.

Note that all the above algorithms employ PD action; however, integral action can be easily incorporated if needed. Also note that the above control approaches allow one to compute a set of generalized forces that will diminish the tracking error. The reaction jet forces and torques and the joint torques can be found by inverting an equation relating generalized forces to actuator forces, i.e. Eq. (3.16).

Having a mathematical model of the system dynamics, developed control laws, and desired trajectories for every output variable, the system performance can now be simulated. This is to be discussed next.

Table 4.1: Comparison of the required computational operations.

Algorithm	Multiplication	Additions
TJ	$3\,N^2$	$3\,N^2 - 2\,N$
MB1	$2\,N^3 + 7\,N^2$	$2\,N^3 + 5\,N^2 - 4\,N$

4.4 Simulation Results

In this section, the performance of the developed model-based algorithms in controlling a multiple manipulator SFFR, is compared to that of the transpose Jacobian algorithm discussed in Section 4.3. The verification procedure of the simulation code is first discussed. Then, the importance of a symmetric vs. a non-symmetric grasp, and the ratio of spacecraft maximum acceleration/deceleration is investigated using a planar example. Next, comparisons between the performance of alternative algorithms during two and three dimensional maneuvers is discussed. It is shown that a simple transpose Jacobian algorithm can yield an acceptable performance, comparable to that of model-based algorithms, with reduced computational burden, which is an important issue in space.

4.4.1 Code Implementation and Verification

The system dynamics model of a multiple manipulator SFFR, which is a central element in the simulation code, has been already verified in a reliable way as explained in Section 3.4.3. The dynamics model in a symbolic (analytical) format is imported to the general simulation routine in FORTRAN, where equations of motion and the developed control laws are integrated, using the Gear algorithm. As expected, applying the MB algorithms under the assumption of exact knowledge of system model and parameters, results in either zero or truncation tracking errors (due to limitations of computational procedures). This is a typical amended result which partly validates the simulation process. Note that in the simulations that follow, effects of model uncertainties are included in the MB laws, by perturbing the mass properties of the model used in the control algorithm with respect to the "true" parameters.

The veracity of the simulation results, has been also investigated by comparing the results for some simple examples to those available in the literature, e.g. motion control of a single arm two-link planar space manipulator in free-flying mode, Papadopoulos and Dubowsky (1991b). The code has been also employed to help an independent research group in Japan (Masutani, Y., Osaka University), eliminating programming oversights of a developed software. Identical results ensures accuracy of the general simulation code for motion control of a multiple manipulator SFFR.

4.4.2 Example 1: Planar Motion

In this section, a planar free-flyer chasing a moving point target, is used to compare and evaluate the control algorithms developed in Section 4.3. The free-flyer includes three open chain appendages, two of which are two-link manipulators, while the third one is a communications antenna, see Figure 4.2.

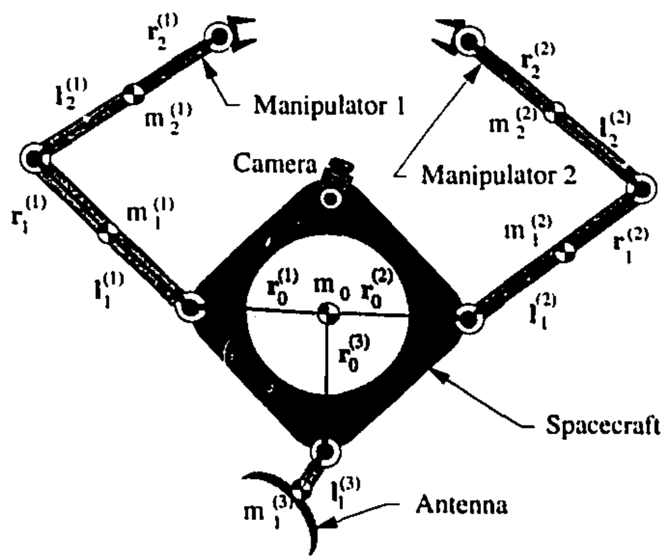


Figure 4.2: A planar three manipulator and appendage free-flyer, Ex. 1.

The spacecraft is equipped with reaction jets which provide the required control forces and torques up to some limited values. The system geometric parameters and mass properties, and the maximum available actuator forces/torques are displayed in Table 4.2. The origin of the inertial frame coincides with the initial position of the system CM, and the vector of generalized coordinates for this 8-DOF system is chosen as

$$\mathbf{q} = [x_{CM}, y_{CM}, \theta_0, \theta_1^{(1)}, \theta_2^{(1)}, \theta_1^{(2)}, \theta_2^{(2)}, \theta_1^{(3)}]^T \tag{4.34a}$$

while the vector of output variables to be controlled is

$$\hat{\mathbf{q}} = [x_0, y_0, \theta_0, x_E^{(1)}, y_E^{(1)}, x_E^{(2)}, y_E^{(2)}, \delta_E^{(3)}]^T \tag{4.34b}$$

Table 4.2-a: Spacecraft parameters and actuator limits, Ex. 1.

$r_0^{(1)}$ (m)	$r_0^{(2)}$ (m)	$r_0^{(3)}$ (m)	m_0 (kg)	I_0 (kg m^2)	$\|F_x\|$ (N)	$\|F_y\|$ (N)	$\|\tau_0\|$ (N-m)
0.5	0.5	0.5	50.0	10.0	20.0	20.0	10.0

Table 4.2-b: Manipulator parameters and joint actuator limits, Ex. 1.

Appendage	i-th body	$r_i^{(m)}$ (m)	$l_i^{(m)}$ (m)	$m_i^{(m)}$ (kg)	$I_i^{(m)}$ (kgm^2)	$\|\tau_i^{(m)}\|$(N-m)
1	1	0.50	0.50	4.0	0.50	7.0
1	2	0.50	0.50	3.0	0.25	5.0
2	1	0.50	0.50	4.0	0.50	7.0
2	2	0.50	0.50	3.0	0.25	5.0
3	1	0.25	0.25	5.0	2.00	7.0

where x_{CM} and y_{CM} are the inertial coordinates of the system CM, x_0 and y_0 are the inertial coordinates of the spacecraft CM, θ_0 is the spacecraft attitude, $\theta_i^{(j)}$ is the i-th joint angle of the j-th manipulator, and $x_E^{(i)}$, $y_E^{(i)}$, and $\delta_E^{(i)}$ are the inertial coordinates and attitude of the i-th end-effector.

It is assumed that the target is in the vicinity of the robotic system, that it is a passive object, i.e. drifting at some constant speed, and that its trajectory is measured by such feedback devices as on-board cameras. Hence, the position and velocity of the target is available in the spacecraft frame.

For the simulation results that follow, the initial values are taken as

$$\mathbf{q}(0) = [0,0,-30\infty,45\infty,90\infty,135\infty,-90\infty,30\infty]^T$$

$${}^0\mathbf{x}_{obj}^0(0) = [3.0,4.0]^T \text{ (m)}$$

$${}^0\mathbf{v}_{obj}^0(0) = [0.05,0.1]^T \text{ (m/s)}$$

$$[x_0(0), y_0(0), \theta_0(0)]^T = [-0.0485\text{m}, -0.0659\text{m}, -\pi/6 \text{ rad}]^T$$

$$[{}^0\dot{x}_0(0), {}^0\dot{y}_0(0), \dot{\theta}_0(0)]^T = (0.01\text{m/s}, 0.01\text{m/s}, 0.001 \text{ rad/s}]^T$$

The final time for the linear motion, t_f, is chosen as 15.0 sec. The planned trajectory for the spacecraft rotation aims to provide a symmetric grasp of the object, by two participating manipulators, to result in minimum disturbances on the spacecraft. Therefore, the final time for the rotational motion is chosen equal to $0.7t_f$, to ensure that the object enters the fixed-base workspace of both end-effectors, approximately at the same time. Taking into account the mass properties of the system and the available thruster forces/torques, the maximum acceleration and deceleration are set to $\mathbf{a}_1 = [0.2, 0.2]^T m/s^2$, $\mathbf{a}_2 = 0.2\mathbf{a}_1$ for the linear motion, and a_1 = 0.05 rad/s^2, a_2= 0.5a_1 for the rotational motion. The importance of symmetric grasp, and of acceleration/deceleration ratio is investigated later on.

Figure 4.3 depicts typical manipulator joint trajectories, and an animated view of the corresponding system maneuver. Note that according to the planned trajectories, the joint angles for the two-link manipulators remain constant during the chase phase (in home configuration), and that they change smoothly during the capture phase (object in manipulator fixed-base workspace). The joint angle for the third appendage (the antenna) changes smoothly so that a fixed inertial orientation is maintained during the maneuver.

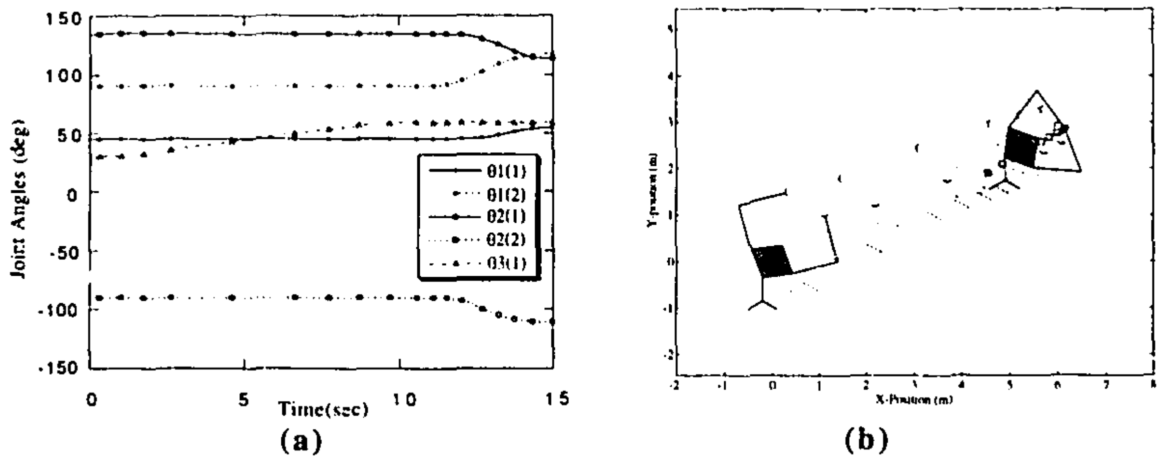


Figure 4.3: (a) Joint angle histories for the two manipulators and the antenna, (b) Animated view of the maneuver.

To include the effects of model uncertainties in the MB laws, the mass properties of the model used in the control circuit were perturbed with respect to the "true" parameters by up to 10%. The gains used for the model-based controllers are

$$\mathbf{K}_p = \text{diag}(70,70,100,100,100,100,100,70)$$

$$\mathbf{K}_d = \text{diag}(15,15,15,15,15,15,15,15)$$

while for the TJ controller these are

$$\mathbf{K}_p = \text{diag}(100,100,80,80,80,80,80,80)$$

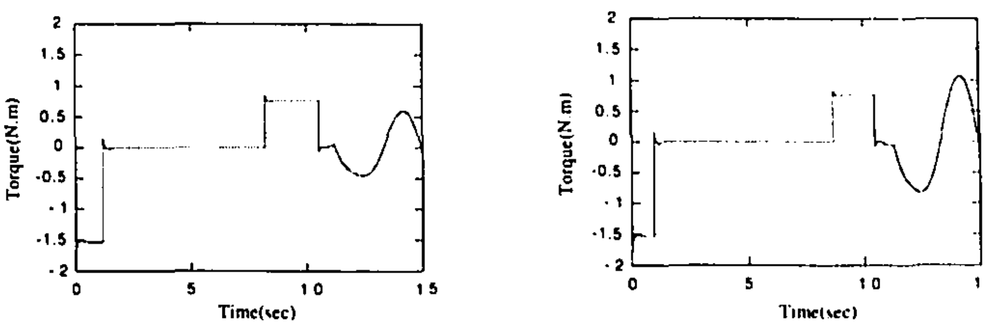
$$\mathbf{K}_d = \text{diag}(150,150,100,100,100,100,100,100)$$

The gain selection for the model-based control was based on error equation settling time and damping criteria, while for the TJ control a heuristic approach was used.

Before going through comparisons between the model-based and TJ algorithms, the importance of symmetric grasp, and the ratio of acceleration/deceleration is investigated by simulation. To this end, the MB1 algorithm as described above is used.

4.4.2.1 Symmetric vs. Non-Symmetric Grasp

Figure 4.4, shows the profile of applied external torque on the spacecraft for different grasp strategies, i.e. (a) symmetric and, (b) non-symmetric grasps. In Figure 4.4(a), i.e. symmetric grasp, the final orientation is chosen so that the axis of symmetry for the spacecraft is aligned with the direction of the object motion, while in Figure 4.4(b) a misalignment of 5.0° between these directions is allowed.



(a) (b)

Figure 4.4: Applied torque on the spacecraft, (a) Symmetric grasp, (b) Non-symmetric grasp.

As it is seen, during the capture phase, $11.0 < t < 15.0$, the torque peak for symmetric grasp is almost half of the one for non-symmetric grasp. Therefore, it can be concluded that a symmetric grasp reduces disturbances on the spacecraft.

4.4.2.2 Maximum Desired Acceleration and Deceleration of the Spacecraft

As discussed earlier, there are two main reasons for choosing the maximum deceleration to be less than the maximum acceleration for a given maneuver duration and on-off thrusting. First, a longer deceleration period results in less thrusting before a grasp, and in less vibration in flexible components like solar panels, and therefore disturbances to the object are reduced. Second, longer deceleration period increases the time available to manipulator motion which results in smoother operation.

Figure 4.5, demonstrates some consequences of the above choice, by comparing a case where $\mathbf{a}_2 = 0.2\mathbf{a}_1$ to one where $\mathbf{a}_2 = \mathbf{a}_1$. As shown in part (a), the former results in lower thruster forces before the grasp, and therefore results in a smaller object disturbance. Also, when $\mathbf{a}_2 = 0.2\mathbf{a}_1$, lower torque on the spacecraft is required to track the desired trajectory, see Figure 4.5(b). In addition, since $\mathbf{a}_2 = 0.2\mathbf{a}_1$ provides a longer duration for manipulators to catch the object, tracking errors are reduced almost 50% with respect to the ones of $\mathbf{a}_2 = \mathbf{a}_1$, Figure 4.5(c).

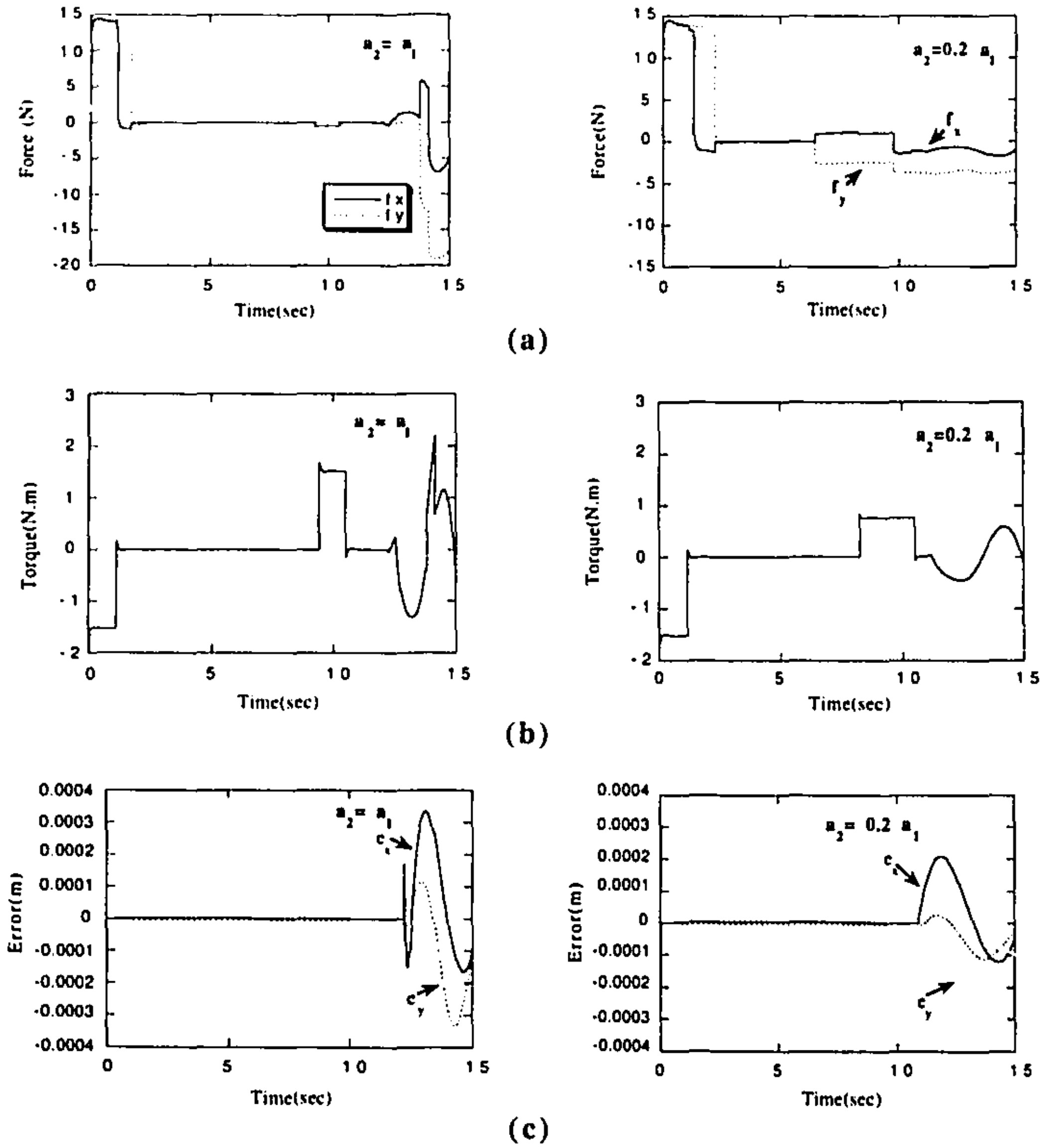


Figure 4.5: The effect of acceleration/deceleration ratio, (a) Spacecraft thruster forces, (b) Applied torque on the spacecraft, (c) First end-effector positioning error.

Next, performance comparisons between the MB and TJ algorithms are presented.

4.4.2.3 Application of Alternative Control Algorithms

For a planar system, the two model-based control algorithms (MB1, MB2), yield almost identical results, and so only the obtained results corresponding to the first control law are presented here. The comparison between these two in a 3-dimensional maneuver is discussed in Section 4.4.3.

Figure 4.6 can be used to compare and evaluate the performance of model-based and transpose Jacobian algorithms. Figure 4.6(a) displays the tracking error for the first manipulator end-effector in the task space. During the chase phase ($0< t <11$), this error is almost zero for MB1, as the manipulators are kept fixed at their home positions (joint-space control phase). When the object enters the manipulator workspace, the manipulators start moving, and tracking errors appear due to dynamic coupling and to transition to the task-space control phase. Note that in the absence of parameter uncertainties, i.e. for perfect model-based control, feedback linearization results in zero tracking errors, as discussed before. However, as it is seen, the performance deteriorates if model uncertainties exist. These errors decrease with time and eventually vanish, in both MB and TJ algorithms.

Comparison of the maximum values of the tracking errors for the two algorithms shows that the errors occurring with TJ are about forty times larger than the errors with MB1, Figure 4.6(a). However, their absolute magnitude may be considered small enough for many space tasks. Comparison of the spacecraft thruster forces, shows that the required forces are about the same for both algorithms, Figure 4.6(b). However, in most parts of the maneuver, the profile for the MB algorithm is staircase, while TJ does not result in such a profile. This is because the TJ algorithm does not use any knowledge of the dynamical behavior of the system. The required joint torques are lower in MB1, see Figure 4.6(c). The variation of the applied joint torques follows the variation of the spacecraft's attitude and tracking errors, which are due to the same reasons, as above.

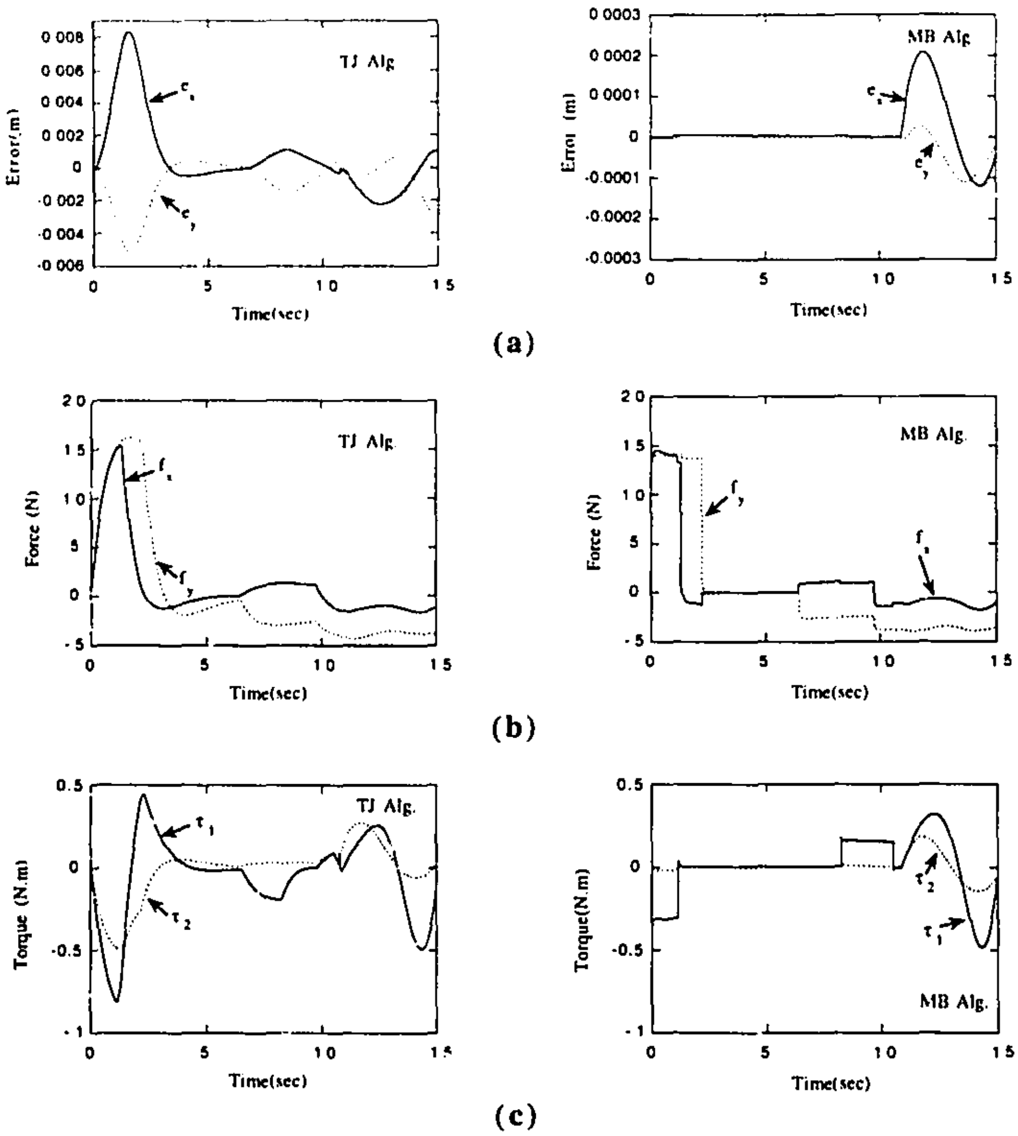


Figure 4.6: TJ compared to MB Control. (a) Tracking position errors for the first end-effector, (b) Thruster forces on the spacecraft, (c) Joint torques for the first arm.

As shown by simulation, model-based algorithms result in smaller errors and lower required torques, as long as model uncertainties are limited. Since torques are lower, supplying less amount of energy would be required, resulting in reduced system weight or longer operation life, important issues in space. However, implementing a model-based control requires increased computational burden, which may not be available. On the other hand, TJ control yields acceptable results (in terms of small errors and reasonable required forces/torques) for executing many tasks in space without requiring knowledge of system dynamics. Therefore, it can be suggested as a good control algorithm candidate, especially when lower computational effort is desired. To support these conclusions further, the developed control algorithms are compared and evaluated by simulating the system performance in a general spatial maneuver. This is to be discussed next.

4.4.3 Example 2: Three-Dimensional Maneuver

In this section, the developed control algorithms are compared and evaluated by simulating the performance of a 3-D free-flyer robot, chasing a moving point target in 3-dimensional space. The total DOF for the simulated system is 14. The spacecraft includes three open chain appendages, two of which are three-DOF manipulators, while the third is a two-DOF communication antenna. The system is equipped with reaction jets on the base, which provide the required control forces and torques up to some limited values. Figure 4.7 shows the system general configuration.

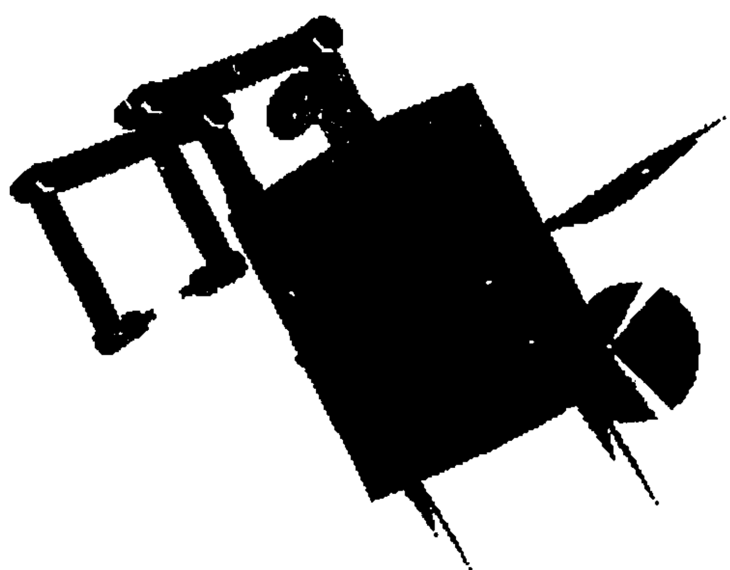


Figure 4.7: A three manipulator and appendage free-flyer, Ex. 2.

4.4.3.1 System Description

The system geometric parameters (according to the nomenclature depicted in Figure 2.2) and mass properties, and also the maximum available actuator forces/torques are displayed in Table 4.3. It should be mentioned that in these tables, all components are given in the corresponding body-fixed frame $\mathbf{xyz}_i$. Each frame is parallel to the principal axes of the corresponding body, and the angle between the z-axis of a frame and the one of the previous frame, according to the D-H convention, is also given in Table 4.3. The origin of the inertial frame coincides with the initial position of the spacecraft CM, which is also defined as the origin of spacecraft body-fixed frame.

The vector of generalized coordinates for this 14-DOF system is selected as follows

$$\mathbf{q} = [x_0, y_0, z_0, \alpha_0, \beta_0, \gamma_0, \theta_1^{(1)}, \theta_2^{(1)}, \theta_3^{(1)}, \theta_1^{(2)}, \theta_2^{(2)}, \theta_3^{(2)}, \theta_1^{(3)}, \theta_2^{(3)}]^T \tag{4.35}$$

while the vector of variables to be controlled is

$$\hat{\mathbf{q}} = [x_0, y_0, z_0, \alpha_0, \beta_0, \gamma_0, x_E^{(1)}, y_E^{(1)}, z_E^{(1)}, x_E^{(2)}, y_E^{(2)}, z_E^{(2)}, \alpha_E^{(3)}, \beta_E^{(3)}]^T \tag{4.36}$$

Table 4.3-a: Spacecraft parameters and actuator limits, Ex. 2.

$r_{0\ x,y,z}^{(1)}$ (m)	$r_{0\ x,y,z}^{(2)}$ (m)	$r_{0\ x,y,z}^{(3)}$ (m)	m_0 (kg)	$I_{0\ xx,yy,zz}$ (kg m²)	$\lvert{}^0\mathbf{f}_s\rvert_{max}$ (N)	$\lvert{}^0\mathbf{n}_s\rvert_{max}$ (N-m)
0.3,-0.2,0.5	-0.3,-0.2,0.5	0,0.3,-0.4	300.0	8.5,10.25,6.25	25,25,25	10,10,10

Table 4.3-b: Manipulator parameters and the joint actuator limits, Ex. 2.

m	i	$\alpha_i^{(m)}$ (deg)	$r_{i\ x,y,z}^{(m)}$ (m)	$l_{i\ x,y,z}^{(m)}$ (m)	$m_i^{(m)}$ (kg)	$I_{i\ xx,yy,zz}^{(m)}$ (kg m²)	$\lvert\tau_i^{(m)}\rvert_{max}$ (N-m)
1	1	0.0	0.0,0.0,0.15	0.0,0.0,-0.15	8.0	0.07,0.07,0.02	10.0
1	2	90.0	0.35,0.0,0.0	-0.35,0.0,0.0	12.0	0.03,0.51,0.51	7.0
1	3	0.0	0.25,0.0,0.0	-0.25,0.0,0.0	10.0	0.03,0.22,0.22	5.0
2	1	180.0	0.0,0.0,-0.15	0.0,0.0,0.15	8.0	0.07,0.07,0.02	10.0
2	2	90.0	0.35,0.0,0.0	-0.35,0.0,0.0	12.0	0.03,0.51,0.51	7.0
2	3	0.0	0.25,0.0,0.0	-0.25,0.0,0.0	10.0	0.03,0.22,0.22	5.0
3	1	-90.0	0.0,0.0,0.15	0.0,0.0,-0.15	3.0	0.03,0.03,0.01	3.0
3	2	90.0	0.20,0.0,0.0	-0.20,0.0,0.0	2.0	0.08,0.08,0.08	3.0

where (x_0, y_0, z_0) denotes the inertial position of the spacecraft CM, $(\alpha_0, \beta_0, \gamma_0)$ is an Euler angle description for the spacecraft attitude, $\theta_i^{(j)}$ is the i-th joint angle of the j-th manipulator, $(x_E^{(i)}, y_E^{(i)}, z_E^{(i)})$ are the inertial coordinates of the i-th end-effector, and $(\alpha_E^{(i)}, \beta_E^{(i)}, \gamma_E^{(i)})$ is an Euler angle description for the i-th end-effector inertial attitude. To implement the third control algorithm, MB2, a vector of generalized speeds similar to $\dot{\mathbf{q}}$ is defined, in which the rate of $(\alpha_0, \beta_0, \gamma_0)$ is substituted by ${}^0\boldsymbol{\omega}_0$.

Note that since the third end-effector is an axisymmetric antenna, only two of the corresponding Euler angles are controlled. These angles and their rates, have to be computed in terms of generalized coordinates and velocities. To this end, the inertial rotation matrix, which relates the end-effector frame to the inertial one, is written in terms of spacecraft attitude parameters and corresponding joint angles, and on the other hand in terms of Euler angles. Setting the two rotation matrices equal and using inverse kinematic relationships yields antenna's Euler angles in terms of the generalized coordinates. Then, expressing the angular velocity of this end-effector in terms of the spacecraft angular velocity and the corresponding joint rates, and also in terms of Euler rates, the relationship between these rates and the vector of generalized speeds can also be obtained.

As discussed earlier, Euler angle model-based control algorithm (MB1) presents the inconvenience of representational singularities. In other words, the inversion of the relation between angular velocity and Euler rates, which is required to find actuator forces/torques based on the control command, is not possible at some orientations where $\mathbf{S}_0$ becomes singular, see Section 4.3. Figure 4.8 shows the errors in spacecraft orientation described by Euler angles, where the system encounters such a non-physical singularity along the planned trajectory, if controlled under MB1 law. To be able to compare the performance of MB1 to that of MB2, tracking the same desired trajectory, the occurrence of such singularities is avoided in the following simulation. This is done by appropriate selection of initial and final values.

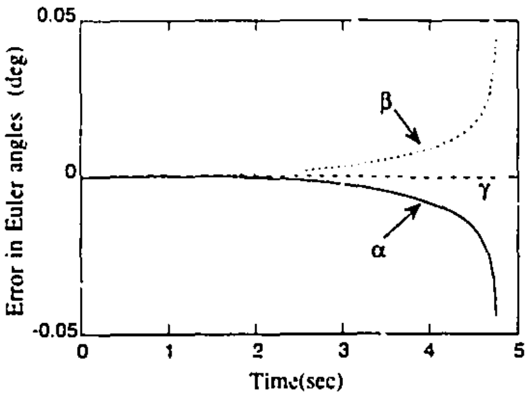


Figure 4.8: Errors in spacecraft orientation encountering a non-physical singularity at time = 4.75 sec.

For the simulation results that follow, the object initial position and relative velocity is taken as ${}^0\mathbf{x}^0_{obj}(0)$ = $[5.0,-6.0,3.0]^T m$, and ${}^0\mathbf{v}^0_{obj}(0)$ = $[0.05,-0.1,0.05]^T$ *m/s*, respectively. The initial position of the spacecraft in the inertial frame is $[x_0(0), y_0(0), z_0(0)]^T$ = $[0.0,0.0,0.0]^T m$, and its orientation $[\alpha_0(0), \beta_0(0), \gamma_0(0)]^T$ = $[0.0, \pi/3, \pi/6]^T$ rad. The spacecraft initial linear velocity is $[\dot{x}_0(0), \dot{y}_0(0), \dot{z}_0(0)]^T$ = $[0.01, 0.01, 0.01]^T m/s$, and its angular velocity is zero. Taking into account the mass properties of the system and the available thruster forces/torques, the maximum acceleration and deceleration are set to $\mathbf{a}_{1_{lin}}$ = $[0.05,0.05,0.05]^T$ m/s^2, and $\mathbf{a}_{2_{lin}} = 0.4\mathbf{a}_{1_{lin}}$ for the linear motion, and $\mathbf{a}_{1_{ang}}$ = $[0.02, 0.02, 0.02]^T$ rad/s^2, and $\mathbf{a}_{2_{ang}} = 0.5\,\mathbf{a}_{1_{ang}}$ for the rotational motion. Finally, the vector of generalized coordinates at initial time is $\mathbf{q}(0)$ = [0,0,0,0∞,60∞,30∞,-90∞,0∞,-90∞,-90∞,180∞,-90∞,30∞,60∞]^T. Figure 4.9 shows the desired path for the spacecraft center of mass and manipulators end-effectors.

To include the effects of model uncertainties in the MB laws, the mass properties of the model used in the control circuit were perturbed with respect to the "true" parameters by up to 30%. The gains used for the MB controllers are $\mathbf{K}_p$=diag(80,...,80,50,50), and $\mathbf{K}_d$=diag(150,...,150,100,100), while for the TJ controller the gains are $\mathbf{K}_p$=diag(300, 300,300,200,...,200,100,100), and $\mathbf{K}_d$=diag(600,600,600,400,...,400,200, 200). The gain selection for the model-based control was based on error equation settling time and damping criteria, while for the TJ control on heuristics. Next, comparisons between MB1, MB2, and TJ algorithms, based on obtained simulation results are discussed.

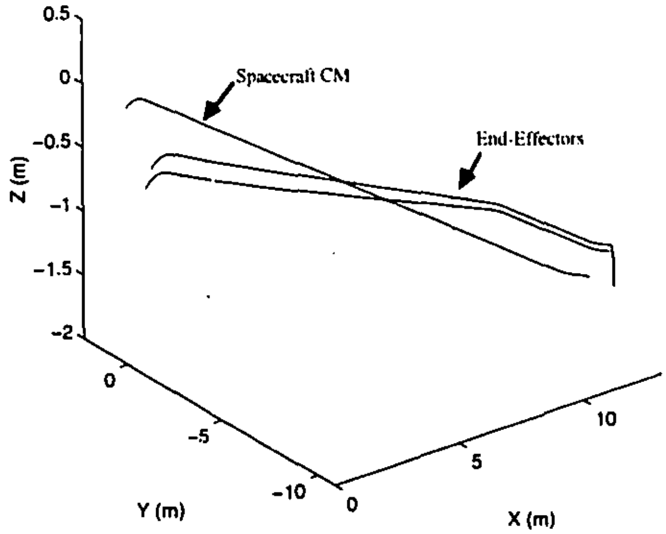


Figure 4.9: The desired path for the spacecraft center of mass and the two end-effectors.

4.4.3.2 Comparison and Discussion

Figures 4.10 to 4.12 can be used to compare and evaluate performance of MB1, MB2, and TJ algorithms. Tracking error for the position of the first manipulator end-effector is shown in Figure 4.10. Other tracking errors (e.g. spacecraft CM position, second manipulator end-effector, etc.) behave similarly. So, Figure 4.10 represents typical error characteristics of the implemented algorithms.

During the chase phase ($0< t <58$), the error for MB algorithms is almost zero, as the manipulators are kept fixed at their home configurations and the whole system moves like a single rigid body. However, for the TJ algorithm, the error is considerable at the beginning of this phase, where the system is accelerating (i.e. $0< t <7$ sec). This is due to the fact that the TJ algorithm is unaware of the dynamical coupling of the system, as it does not include dynamics terms in its structure. When the object enters the manipulator workspace, the manipulators start moving, and some tracking errors appear due to the dynamic coupling and also transition from joint-space to task-space control phase. In all three algorithms, given enough time, these errors decrease and eventually vanish. Comparison of the maximum

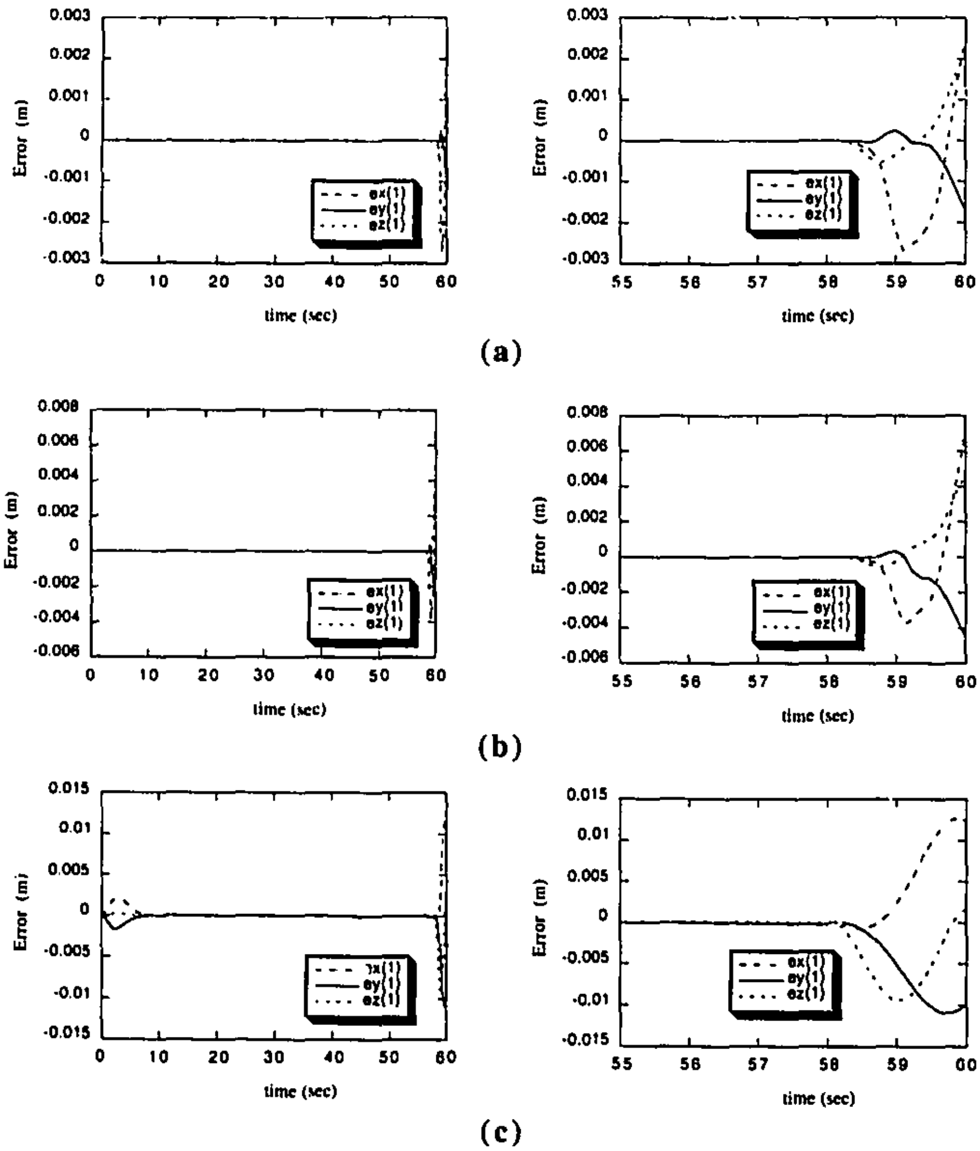


Figure 4.10: Tracking position errors for the first end-effector, (a) Model-Based Control, MB1. (b) Model-Based Control, MB2. (c) Transpose Jacobian Control.

maximum tracking errors for these algorithms shows that the errors occurring with the TJ are about two-five times higher than the errors with the MB algorithms[7], although their absolute magnitude may be considered small enough for performing a wide range of tasks.

Figure 4.11 shows the applied control forces on the spacecraft. Comparison of the spacecraft thruster forces, shows that the required forces are about the same for all three algorithms. However, for most maneuver segments, the profile is staircase for the MB algorithms (which is easier to follow, in practical implementations), but not for the TJ control. Again, this is because the TJ algorithm does not take into account the dynamical behavior of the system.

Note that due to dynamic coupling, the rotational deceleration requires an additional application of thruster forces, so that the translational motion continues to track. This occurs near the end of the attitude maneuver at the 45-th sec of the motion, and can be recognized in all three plots (Figure 4.11), circled in part (c).

Figures 4.12 displays applied torques to control the spacecraft attitude and motion of the first manipulator, near the end of the maneuver ($53.0 < t < 60.0$). In general, variation of the applied torques follows the variation of tracking errors, and is due to the same reasons, as above. As it is seen, the required torques are almost the same for all three algorithms, though MB2 is less demanding. Note that the profile of a component of applied torques on the spacecraft only touches the saturation limit (10 N-m) for MB2, while for the others it remains at that limit for a relatively long time. Also, it should be noted that the joint torques for the TJ algorithm are about 20-60% off compared to those of the MB1 and MB2. Finally, comparing part (a) with part (c) of Figures 4.11 and 4.12, it is interesting to note that profile of (c) is a smooth approximation of the profile of (a). Clearly, this is because the TJ algorithm is an approximation of MB1 and so are the control forces/torques.

7- Note that to include the effects of model uncertainties in the MB laws, the mass properties of the model used in the control law were perturbed with respect to the "true" parameters by up to 10% in Ex. 1, and by up to 30% in Ex. 2. As expected and shown by simulation, the larger these uncertainties are the worse tracking is.

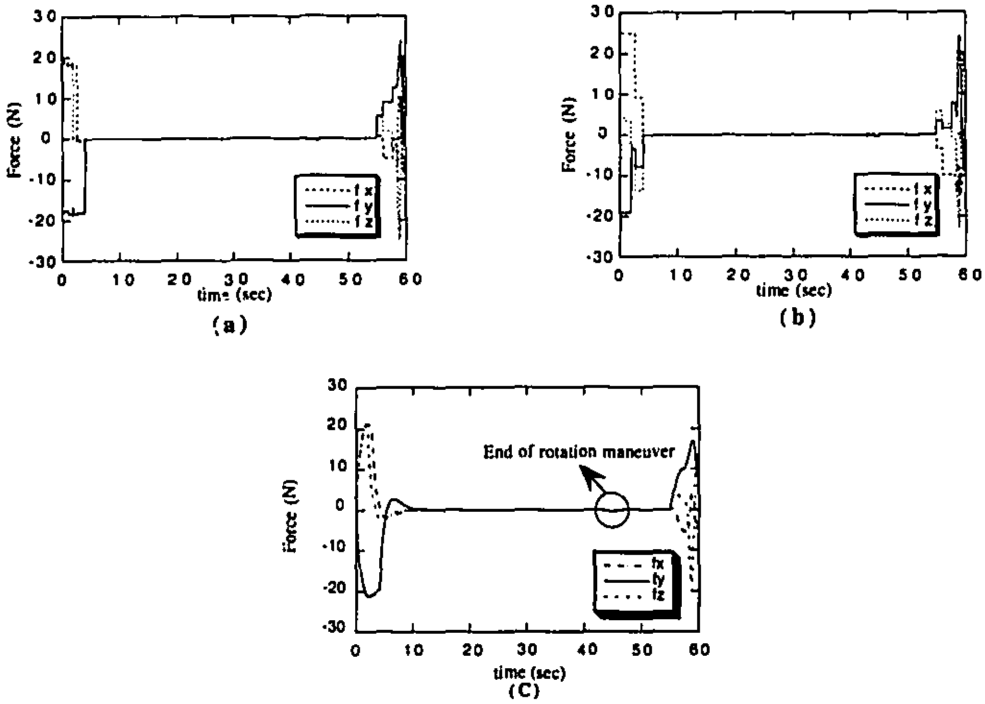


Figure 4.11: Thruster forces on the spacecraft, (a) MB1, (b) MB2, (c) TJ, algorithm.

As this general 3-dimensional maneuver reveals, consistent to the previous example in planar motion, the MB algorithms result in a better tracking and smaller errors, even in the presence of model uncertainties. The MB2 controller is preferred because as shown in the development of this algorithm (see Section 4.3.1.2), it overcomes the non-physical singularity problem. However, implementing a model-based control requires increased computational burden. The TJ control, with relatively high gains, yields acceptable results and can be considered as a good control algorithm candidate, especially when low computational costs are required. However, due to the presence of noise and unmodelled dynamics, the use of very high gains will be limited in practice. These results motivate further work on the TJ algorithm, aiming at overcoming the requirement of larger gains and consequently sensitivity to noise, and the lack of information about the dynamics of the system, a problem which appears more clearly in tracking fast trajectories.

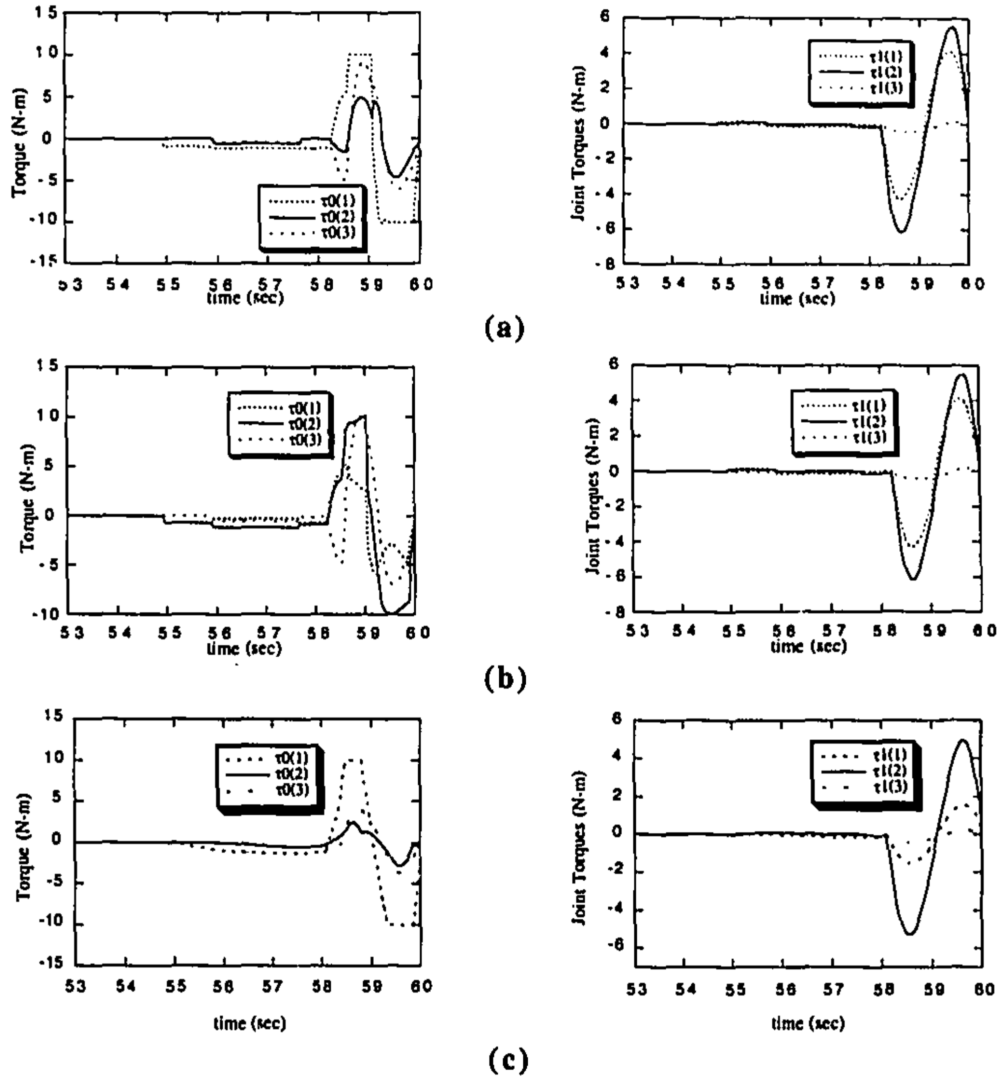


Figure 4.12: Applied torques on the spacecraft (left) and joint torques for the first end-effector (right), (a) MB1, (b) MB2, (c) TJ Algorithm.

4.5 Summary and Conclusions

In this chapter, coordination between a spacecraft motion and its several end-effectors to capture a moving space object, was investigated. Taking into account the object motion relative to the spacecraft, as well as thruster and actuator saturation limits, appropriate trajectories for the spacecraft and its manipulators motion were planned. Two model-based algorithms, and a transpose Jacobian control algorithm were developed. The Euler angle model-based control algorithm (MB1) presents the inconvenience of representational singularities at some orientations. To overcome this problem, an Euler parameter model-based control algorithm was proposed as the second model-based control algorithm (MB2).

As shown by simulation, a symmetric grasp reduces disturbances on the spacecraft. Also, choosing the maximum deceleration to be less than the maximum acceleration for a given maneuver duration results in a smoother operation. It was shown that the model-based algorithms result in smaller errors, as long as model uncertainties are limited. However, due to the complexity of space robotic systems, the performance of these algorithms deteriorates if higher levels of model uncertainties exist. Also, implementing a model-based control requires increased computational burden, which may not be available. On the other hand, the TJ algorithm with relatively high gains, yields acceptable results (in terms of small errors and reasonable required forces/torques) for executing many tasks in space, without requiring knowledge of system dynamics. Therefore, this simpler algorithm controller as an approximation of the MB1, can be considered as a good candidate especially when lower computational power is available.

Note that the use of very high gains for the TJ algorithm will be limited due to the presence of noise and unmodelled dynamics in practice. Also, the lack of information about the system dynamics, causes poor performance of the algorithm in tracking fast trajectories. Therefore, further work on the TJ algorithm, to improve its characteristics, is required. This is discussed in the next chapter.

Chapter 5

Modified Transpose Jacobian Control and its Application to Space Robotics

5.1 Introduction

Transpose Jacobian (TJ) control is one of the simplest algorithms used to control robotic manipulators. As shown previously, the TJ algorithm with relatively high gains, results in acceptable tracking performance of space free-flyers, without requiring knowledge of system dynamics. Therefore, it is a good control algorithm candidate, especially when lower computational efforts are required. However, since it is not dynamics-based, poor performance may result in tracking of fast trajectories. Use of high gains can deteriorate performance seriously in the presence of feedback measurement noise. Another drawback is that there is no formal method of selecting its control gains.

In this chapter, a new Modified Transpose Jacobian (MTJ) algorithm is presented which employs stored data of the control command in the previous time step, to yield an improved performance. In fact, the MTJ algorithm as developed in Section 5.2, is based on an approximation of feedback linearization methods, without requiring a priori knowledge of plant dynamics. The gains of this new algorithm can be selected more systematically, and do not need to be large, hence the noise rejection characteristics of the algorithm are improved.

In Section 5.3, simulation results are presented which compare tracking performance of the MTJ algorithm to that of the TJ and Model-Based (MB) algorithms. To focus on algorithmic aspects, a simple two link planar manipulator is first simulated. Then, the new MTJ algorithm is applied to the coordinated motion control of a 14-DOF space free-flying robotic system. Results show that tracking performance of this new algorithm is comparable to that of Model-Based algorithms, without requiring a priori knowledge of plant dynamics, and with reduced computational burden. Therefore, this new MTJ algorithm is a good candidate for controlling multi-DOF space robots, especially where computational power is limited.

5.2 MTJ Control Law

5.2.1 Motivation

As discussed before, using the expressions for the kinetic and potential energy, and applying Lagrange's equations for a robotic system, the dynamics model can be obtained as

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{Q}(\mathbf{q}) \tag{5.1}$$

where all gravity and nonlinear velocity terms are contained in vector **C**. Gravity terms are practically zero in microgravity environments, and therefore can be neglected in the design of control laws for space robots. In terrestrial applications, these terms may cause static positioning errors in control, and in such case, they must be compensated separately. Therefore, it is assumed that the vector **C** contains only nonlinear velocity terms.

The output speeds, $\dot{\hat{\mathbf{q}}}$, associated with the output variables to be controlled, $\hat{\mathbf{q}}$, are obtained from the generalized speeds $\dot{\mathbf{q}}$ using a Jacobian matrix, $\mathbf{J}_C$, as

$$\dot{\hat{\mathbf{q}}} = \mathbf{J}_C(\mathbf{q})\,\dot{\mathbf{q}} \tag{5.2}$$

Assuming that this Jacobian matrix is square and non-singular, Eq. (5.1) can be written in terms of the output variables as follows

$$\hat{\mathbf{H}}(\mathbf{q})\ddot{\hat{\mathbf{q}}} + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}}) = \hat{\mathbf{Q}}(\mathbf{q}) \tag{5.3}$$

where $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{Q}}$ can be obtained according to Eqs. (4.10).

To control such a system, a Model-Based (Computed Torque) control law such as

$$\mathbf{Q} = \mathbf{J}_C^T \left\{ \hat{\mathbf{H}} \left[\mathbf{K}_p \mathbf{e} + \mathbf{K}_d \dot{\mathbf{e}} + \ddot{\hat{\mathbf{q}}}_{des} \right] + \hat{\mathbf{C}} \right\} \tag{5.4}$$

can be applied. This law linearizes and decouples the system equations to a set of second order differential equations

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{0} \tag{5.5}$$

where $\mathbf{K}_p$, and $\mathbf{K}_d$ are positive definite gain matrices, and **e** is the tracking error defined as

$$\mathbf{e} = \hat{\mathbf{q}}_{des} - \hat{\mathbf{q}} \tag{5.6}$$

Under the usual assumption of known system dynamics structure, and known geometric and mass properties, the control law given by Eq. (5.4) guarantees asymptotic convergence of the tracking error to zero. However, if these assumptions are violated, the error may never converge. In addition, this control law requires a significant computational effort[8] which may not be available on a space system.

As discussed in the previous chapter[9], if high enough gains are used, the control law of Eq. (5.4) can be approximated by the simple Transpose Jacobian (TJ) controller as

$$\mathbf{Q} = \mathbf{J}_C^T \{ \mathbf{K}_p \mathbf{e} + \mathbf{K}_d \dot{\mathbf{e}} \} \tag{5.7}$$

8- To apply a Model-Based (Computed Torque) control law, $\hat{\mathbf{H}}$ and $\hat{\mathbf{C}}$ have to be computed. Considering Eqs. (4.5), it can be seen that computation of $\hat{\mathbf{H}}$ and $\hat{\mathbf{C}}$ requires inversion of the Jacobian matrix and calculation of its time derivative which depending on the system degrees-of-freedom may be quite cumbersome. The number of matrix multiplications in obtaining these expressions, is also considerable. The required computational operations can be seen in Table 4.1, though the assumptions made in preparation of this table exclude the operations for inverting the Jacobian matrix and calculating its time derivative.

9- See Section 4.2.2.

which does not require a priori knowledge of the system dynamics. Besides simplicity, an advantage of this algorithm is that if a physical singularity is encountered, the controller given by Eq. (5.7) may result in errors but will not fail computationally. The action of this controller can be understood by imagining generalized springs and dampers, along the variables under control, connected between the corresponding body and the desired trajectories; the stiffer the gains are, the better the tracking should be. However, due to the presence of noise and unmodelled dynamics, the use of very high gains is limited in practice. Note that computation of $\hat{\mathbf{Q}}$ based on Eqs. (5.7) and (4.5), does not result in the error dynamics given by Eq. (5.6), anymore.

The advantages of using the TJ controller motivate further work on this algorithm, aiming at improving its performance and limiting its drawbacks.

5.2.2 Derivation of MTJ Control Law

To achieve both precision and simplicity, the TJ control law defined by Eq. (5.7) is now modified, to approximate a feedback linearization solution, as

$$\mathbf{Q} = \mathbf{J}_C^T \{ \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + \mathbf{h}(t) \} \tag{5.8}$$

where $\mathbf{h}$(t) is a term to be determined, $\mathbf{K}_p$ and $\mathbf{K}_d$ are positive definite gain matrices, and $\mathbf{e}$ is the tracking error defined in Eq. (5.6). Substitution of Eq. (5.8) into Eq. (5.3), yields

$$\mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \hat{\mathbf{H}} \ddot{\mathbf{q}} + \hat{\mathbf{C}} - \mathbf{h}(t) \tag{5.9}$$

which is equivalent to

$$\mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \hat{\mathbf{Q}} - \mathbf{h}(t) \tag{5.10}$$

It can be seen that if the right hand side (RHS) of Eq. (5.9), becomes equal to zero, then the tracking error converges to zero, and the algorithm works like a Model-Based algorithm, albeit with a simpler implementation. Note that inclusion of the second derivative of the error, $\ddot{\mathbf{e}}$, in Eq. (5.8) results in

$$\mathbf{Q} = \mathbf{J}_q^T \{ \ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + \mathbf{h}(t) \} \tag{5.11}$$

and then

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \hat{\mathbf{H}} \ddot{\mathbf{q}} + \hat{\mathbf{C}} - \mathbf{h}(t) \tag{5.12}$$

which results in an error dynamics similar to that of the MB algorithms, if the RHS of Eq. (5.12) becomes equal to zero. However, inclusion of this signal requires acceleration measurements or an estimator, and may be difficult to obtain in practice.

To make the RHS of Eq. (5.9) or (5.12) be close to zero, Eq. (5.10) suggests that a good approximation can be obtained by taking $\mathbf{h}$(t) equal to $\hat{\mathbf{Q}}$ at a previous small time step, $\hat{\mathbf{Q}}|_{t-\Delta t}$. However, inclusion of this term may result in high joint torque requirements, when relatively high $\mathbf{e}$ or $\dot{\mathbf{e}}$ are imposed due to disturbances. To tackle these disturbances, the standard TJ algorithm can be instantly applied. Therefore, the following form is adapted

$$\mathbf{h}(t) = k\, \hat{\mathbf{Q}}\big|_{t-\Delta t} \tag{5.13}$$

where the *regulating factor*, k, is defined as

$$k = \begin{cases} 0 & when\ |\mathbf{e}| \geq \varepsilon\ or\ |\dot{\mathbf{e}}| \geq \dot{\varepsilon} \\ 1 & when\ |\mathbf{e}| < \varepsilon\ \&\ |\dot{\mathbf{e}}| < \dot{\varepsilon} \end{cases} \tag{5.14}$$

where ε and $\dot{\varepsilon}$ represent sensitivity thresholds. Note that factor k is initially taken equal to zero, resulting in a TJ control law at the first time step. To simplify the on-off switch for factor k, the following continuous expression can be used

$$k = \exp(-(\frac{|\mathbf{e}|}{e_{\max}} + \frac{|\dot{\mathbf{e}}|}{\dot{e}_{\max}})) \tag{5.15a}$$

where $e_{\max}$, and $\dot{e}_{\max}$ are positive real numbers which correspond to another representation of the sensitivity threshold. Note that relatively low values for sensitivity thresholds, would make the algorithm work like the standard TJ control law. In practice, $\mathbf{K}_p$ and $\mathbf{K}_d$ can be chosen as diagonal matrices, and so can be selected the regulating factor. Then, factor k in Eq. (5.13) should be replaced by a diagonal matrix $\mathbf{K}$, where its elements can be defined as

$$k_{ii} = \exp(-(\frac{|e_i|}{e_{\max_i}} + \frac{|\dot{e}_i|}{\dot{e}_{\max_i}})) \tag{5.15b}$$

Including the second term in Eq. (5.15), based on the error first rate, introduces a sense of anticipation, without compromising the smoothness of response. Similarly, one can include another term based on the second rate of error, if available. However, this makes the algorithm more sensitive, and therefore sharp variations of actuator forces/torques may result.

Application of the MTJ algorithm

$$\mathbf{Q} = \mathbf{J}_C^T \{ \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + k \hat{\mathbf{Q}}|_{t-\Delta t} \} \tag{5.16}$$

with proper selection of the sensitivity thresholds (so that the modifying term is reasonably activated) and small time steps, results in the following error equation

$$k_{d_i} \dot{e}_i + k_{p_i} e_i \cong 0 \tag{5.17}$$

where diagonal gain matrices, $\mathbf{K}_p$ and $\mathbf{K}_d$, have been used. Therefore, using Eq. (5.17), the control gains can be selected in a more systematic manner, as their ratio determines error time constant, and their magnitude determines the magnitude of the control command which should be adjusted based on actuator capabilities.

Considering Eq. (5.16), it can be deduced that the MTJ requires $3N^2+N+2$ multiplications, and $3N^2-N+1$ additions. Comparing to the depicted results in Table 4.1, these are almost the same as those for the TJ algorithm, and still significantly less compared to the ones needed for implementing the MB algorithms. Note that it is assumed that the inverse of the Jacobian matrix and its time derivative, which are required for implementing the MB algorithms, are available symbolically, and hence computations involving these are not counted in Table 4.1.

The above analysis reveals the simplicity (concerning a priori knowledge requirement of system dynamics) and efficiency (in terms of the required computational effort) of both the standard TJ and the new MTJ law compared to the MB algorithms. In addition, the MTJ yields approximately linearized error dynamics, and therefore an improved performance over the standard TJ algorithm. Next, based on Lyapunov's theorems, stability analysis of the developed MTJ algorithm is studied.

5.3 Simulations and Comparisons

In this section the performance of the new MTJ control, as given by Eq. (5.16), is evaluated by simulation, and compared to the standard TJ, Eq. (5.7), and model-based (MB) algorithms, Eq. (5.4). First, to focus on algorithmic aspects, a simple two link planar manipulator is simulated. Performing low-speed vs. high-speed tracking task, selection of larger gain for the TJ, and noise rejection characteristics of the considered algorithms are investigated in this Example. Then, the new MTJ algorithm is applied on coordinated motion control of a 14-DOF space free-flying robotic system, and simulation results are compared to those of the other algorithms.

5.3.1 Example 1: Two-Link Fixed-based Manipulator

The simulated system is a simple 2-link planar manipulator on a horizontal plane, see Figure 5.1 (a). The task is tracking a trajectory defined by

$$\begin{aligned} x_{des} &= \sqrt{l_1^2 + l_2^2}\cos(\omega t + \pi/4) + 0.1\sin(5\omega t) \\ y_{des} &= \sqrt{l_1^2 + l_2^2}\sin(\omega t + \pi/4) + 0.1\sin(5\omega t) \end{aligned} \tag{5.18}$$

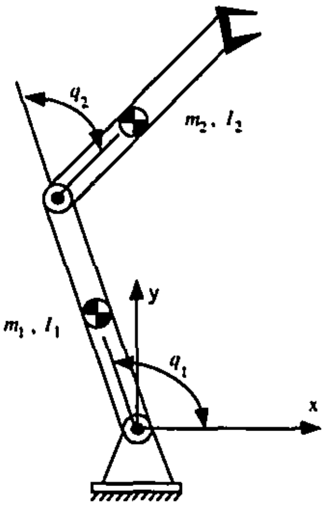
This trajectory corresponds to a perturbed circular path, as shown in Figure 5.1 (b). The motion speed along the path can be selected by setting the cyclical frequency ω.

The mass properties of the system are $m_1 = 4.0\,kg$, $I_1 = 0.333\,kg.m^2$, $m_2 = 3.0\,kg$, and $I_2 = 0.30\,kg.m^2$, and the link lengths are $l_1 = 1\,m$ and $l_2 = 1\,m$. The initial conditions for joint angles and derivatives are

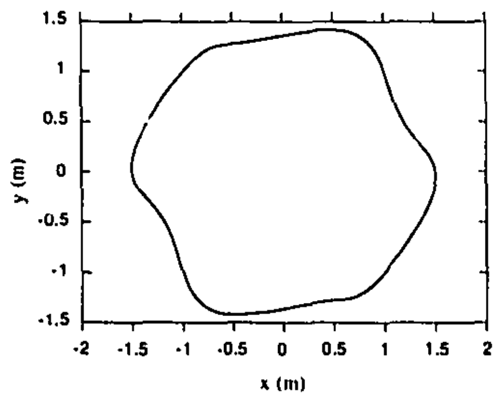
$$\left(q_1(0), q_2(0), \dot{q}_1(0), \dot{q}_2(0)\right) = \left(0.03, \pi/2, 1.5, -1.0\right) \qquad (rad, rad/\sec)$$

which correspond to some initial position and velocity errors.

The sensitivity thresholds for the MTJ algorithm, e_{max} and $\dot{e}_{max}$ in Eq. (5.15a) are taken equal to 1 m and 10 m/sec, respectively. These large values for e_{max} and $\dot{e}_{max}$, yield $k = 1.0$ throughout the whole duration of the simulation after the first time step (which is zero, according to the definition). The time step Δt_i is held constant, and equal to 10.0 msec. To establish a fair comparison, the gains for the algorithms under comparison are selected such that the peaks of the required joint torques are approximately equal. The Gear method for solving differential equations, is used in all simulations.



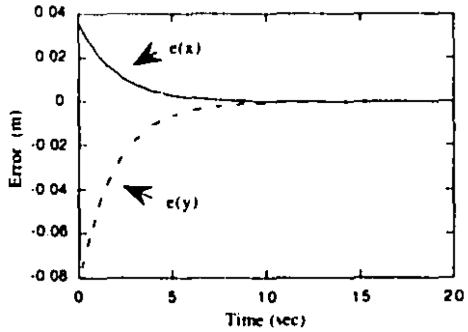
(a)



(b)

Figure 5.1: (a) A two-link planar manipulator, (b) Desired tracking path.

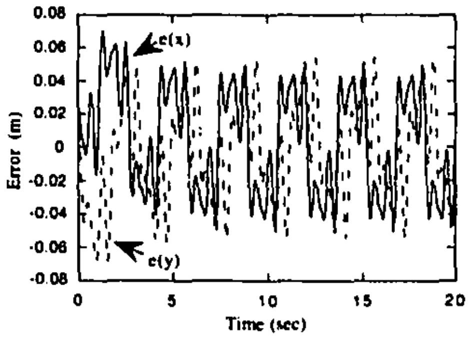
Low-Speed vs. High-Speed Tracking Task. The performance of the TJ and MTJ algorithms, in terms of the end-point error in a low-speed tracking task (ω=0.05 rad/s), is compared in Figure 5.2. For the MTJ algorithm $\mathbf{K}_p$ = diag(30, 30), $\mathbf{K}_d$ = diag(60, 60), while for the TJ algorithm the gains are twice these values. It can be seen that both algorithms result in a fairly similar response. However, errors for the TJ algorithm may increase initially to higher values, before they converge to zero, see for example *e(y)* in Figure 5.2 (a).



(a) (b)

Figure 5.2: Tracking errors for low-speed task, (a) TJ algorithm, (b) MTJ algorithm.

Figure 5.3 shows the end-point tracking error in a high-speed tracking (ω=2.0 rad/s). As shown in this figure, the MTJ algorithm results in smaller tracking errors, and therefore is preferred. This poor performance of the TJ algorithm, is due to the fact that it is not dynamics-based. However, one would expect that by selecting very high gains, its performance can be improved.



(a) (b)

Figure 5.3: Tracking errors for high-speed task, (a) TJ algorithm, (b) MTJ algorithm.

To investigate this possibility, the previous gain values for the MTJ are used, while for the TJ fairly high gains are selected, see Table 5.1. Besides, the task speed is reduced to ω=1.0 rad/s. Here, in addition to the TJ and MTJ algorithms, two cases of model-based (MB) algorithms are also considered. In the first case, it is assumed that the mass properties are completely known, while in the second one, the mass properties of the dynamics model in the controller are perturbed by 10% with respect to the *true* values. For the perfect MB, the chosen gains are fairly low which correspond to a settling time of 2.0 sec, and a damping ratio of 0.7. For the second MB case, these low gains result in relatively large tracking errors, therefore they are selected equal to the ones for the MTJ.

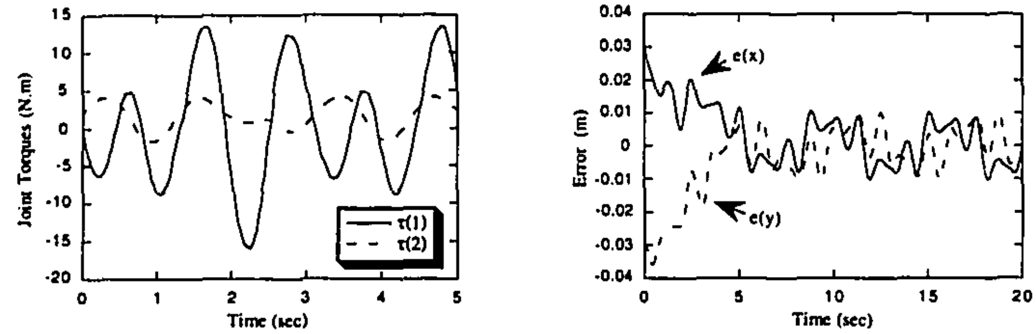
Table 5.1: Selected gains for alternative algorithms, Ex. 1.

Algorithm	$\mathbf{K}_p$	$\mathbf{K}_d$
TJ	diag(150, 150)	diag(300, 300)
MTJ	diag(30, 30)	diag(60, 60)
MB, case 1	diag(8, 8)	diag(4, 4)
MB, case 2	diag(30, 30)	diag(60, 60)

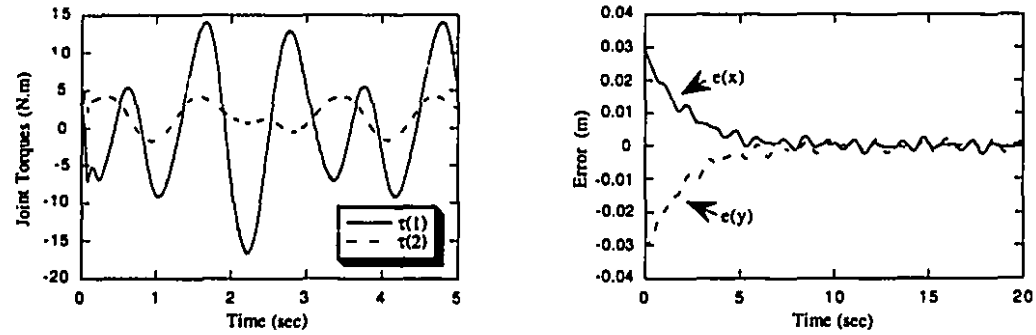
As Figure 5.4 shows, due to properly adjusted gains, the peaks of joint torques for all four algorithms are about the same, which as mentioned before, establishes a fair comparison. Nevertheless, it can be seen that, even with relatively very high gains for the TJ, the resulting tracking errors of the MTJ are still about five times smaller than the ones of the standard TJ, and even better than the ones of the perturbed MB (case 2) algorithm. In other words, the advantage of MB laws is lost if the parameters are not known exactly.

It should be mentioned that the total energy consumption of each algorithm for performing this task, given by the time integral of $\sum_{i=1}^{2} |\tau_i \dot{q}_i|$, is almost the same, i. e. in correspondence to Figure 5.4, (a) 153, (b) 156, (c) 153, and (d) 154 Joule.

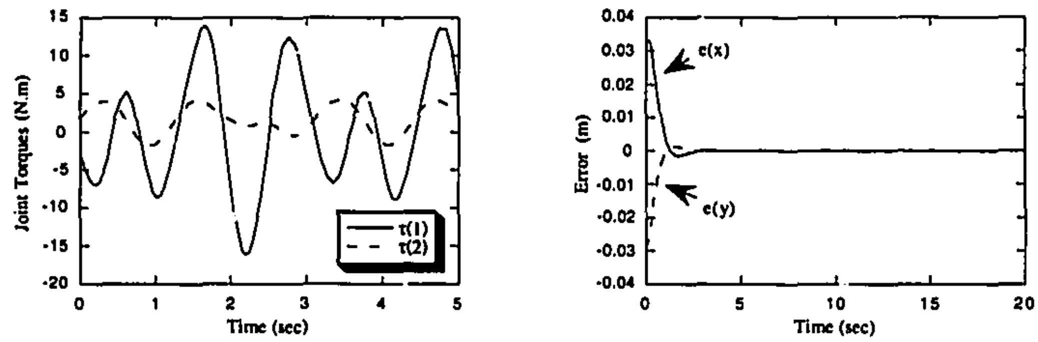
Noise Rejection Characteristics. In practice, noise corrupts any available feedback. Therefore, one should examine the noise rejection capabilities of would be implemented algorithms, especially of those that rely on high gains. The previous simulation is repeated now, assuming that measurements of joint angles and their rates are corrupted by white noise whose amplitude is 2% of the output's magnitude. Although the performance in terms of the average tracking errors is almost the same as before, the variation in the required torques is larger. As shown in Figure 5.5, the required torques for



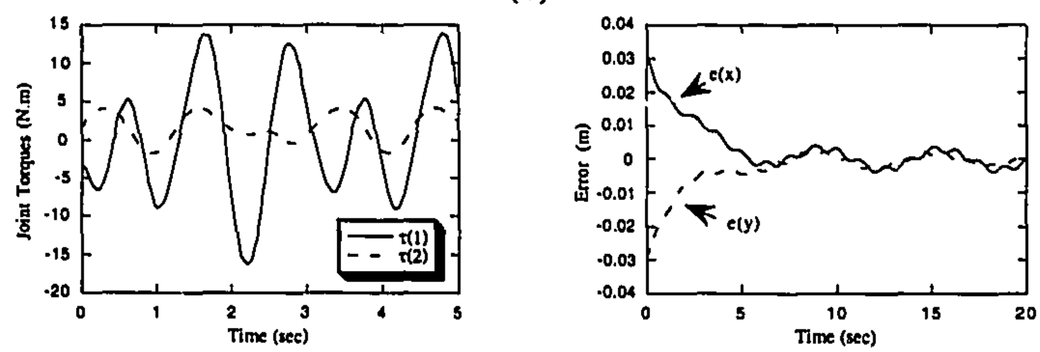
(a)



(b)



(c)



(d)

Figure 5.4: Joint torques and tracking errors, (a) TJ with high gains, (b) MTJ, (c) MB, case 1, (d) MB, case 2.

the MTJ algorithm are almost as smooth as those for a perfect MB control, while the noise rejection characteristics for the TJ algorithm are poorer, due to the high gains employed. Note that for the MB algorithms, having a noisy feedback affects the elements of controller dynamics, which in the presence of uncertainties (the second MB case) as requires larger gains, results in a poor noise rejection characteristics, see Figure 5.5 (d).

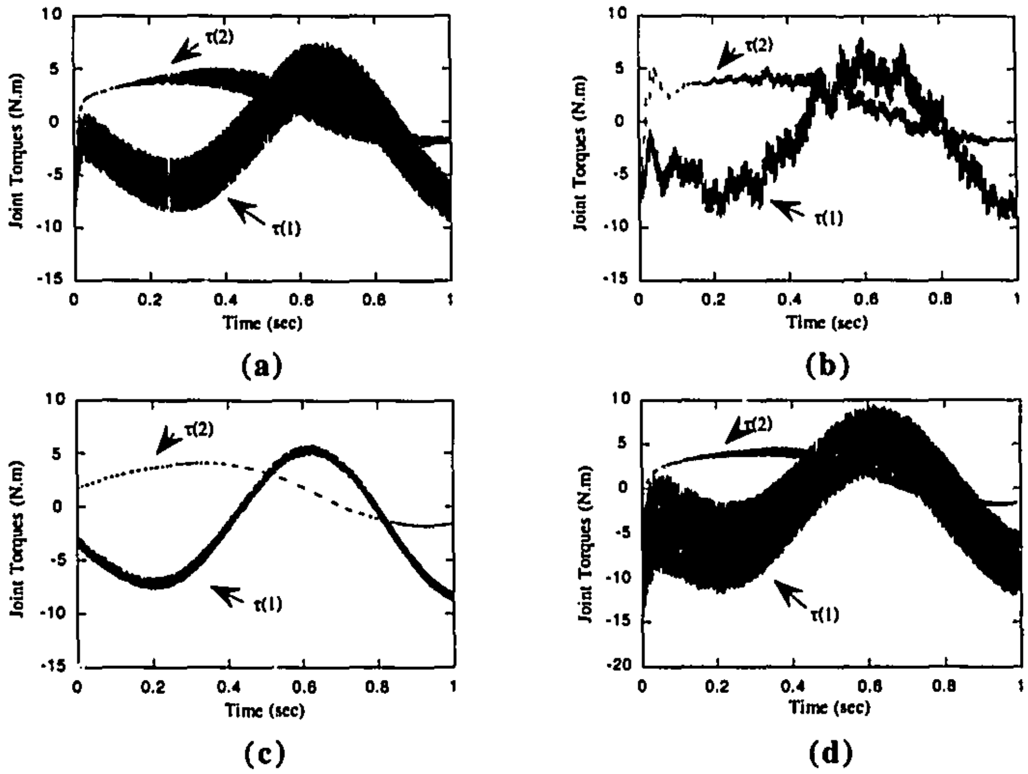


Figure 5.5: Joint torques in the presence of noisy feedback, (a) TJ, high gains, (b) MTJ, (c) MB, case 1, (d) MB, case 2.

It can be concluded that for better tracking, larger gains are required for the TJ algorithm, and these lead to poor noise rejection characteristics. Also, high frequency inputs can excite flexible system modes, and consequently decrease the accuracy, and the useful life of a system. Hence, using high gains is not a viable option. On the other hand, the new MTJ algorithm, by being an approximation of a feedback linearization algorithm, does not require high gains, or a high computational power, while its performance is comparable to that of the Model-Based algorithms.

Next, the new MTJ algorithm is applied to the coordinated motion control of a space free-flying robotic system, and the results are compared to those of the standard TJ and model-based (MB) algorithms.

5.3.2 Example 2: Multiple Arm Space Free-Flying Robotic System

In this section the 14-DOF space free-flyer, described in Section 4.4.3, is simulated in capturing a moving object. The generalized coordinates and the output variables are those already defined in Eqs. (4.35-36). Also, all of the initial values and the required parameters for planning the desired trajectory, are those given in Section 4.4.3.1. Here, the simulated algorithms are

- The MB algorithm, based on Eq. (5.4);
- The standard TJ controller, Eq. (5.7);
- The MTJ algorithm as given in Eq. (5.16), MTJ1;
- The MTJ controller using a second derivative of the error, Eq. (5.11), MTJ2.

To include the effects of model uncertainties in the MB law, the mass properties of the model used in the control algorithm are perturbed with respect to the true parameters by up to 5%. Table 5.2 shows the gains used for alternative controllers. The size of the time step, Δt_n for the MTJ implementation, is held constant and equal to 10.0 msec. The sensitivity thresholds for the MTJ controllers, to be substituted into Eq. (5.15b), are $\mathbf{e}_{max}$ = (1e-2,1e-2 ,1e-2,1e-2,1e-2,1e-2,1e-1,...,1e-1)^T, and $\dot{\mathbf{e}}_{max}$ = (1e-1,1e-1,1e-1,1e-1,1e-1,1e-1,1.0,..., 1.0)^T. The Gear method for solving differential equations, is used in all simulations.

Table 5.2: Selected gains for alternative algorithms, Ex. 2.

Algorithm	$\mathbf{K}_p$	$\mathbf{K}_d$
TJ	diag(300, 300,300, 200,...,200, 100,100)	diag(600,600,600, 400,...,400, 200,200)
MB, MTJ1, MTJ2	diag(150,...,150, 50,50)	diag(300,...,300, 100,100)

Figure 5.6 shows typical tracking position errors for an end-effector. For the TJ algorithm, these errors are much higher (almost 50 times higher than those of the MTJ), especially when the system is accelerating. As discussed before, this is because the TJ algorithm is unaware of the dynamical behavior of the system. However, it is seen that the error for the MTJ algorithm (both MTJ1 and MTJ2) remains very small, throughout the maneuver. Note that here, the MTJ1 and MTJ2, result in similar tracking errors.

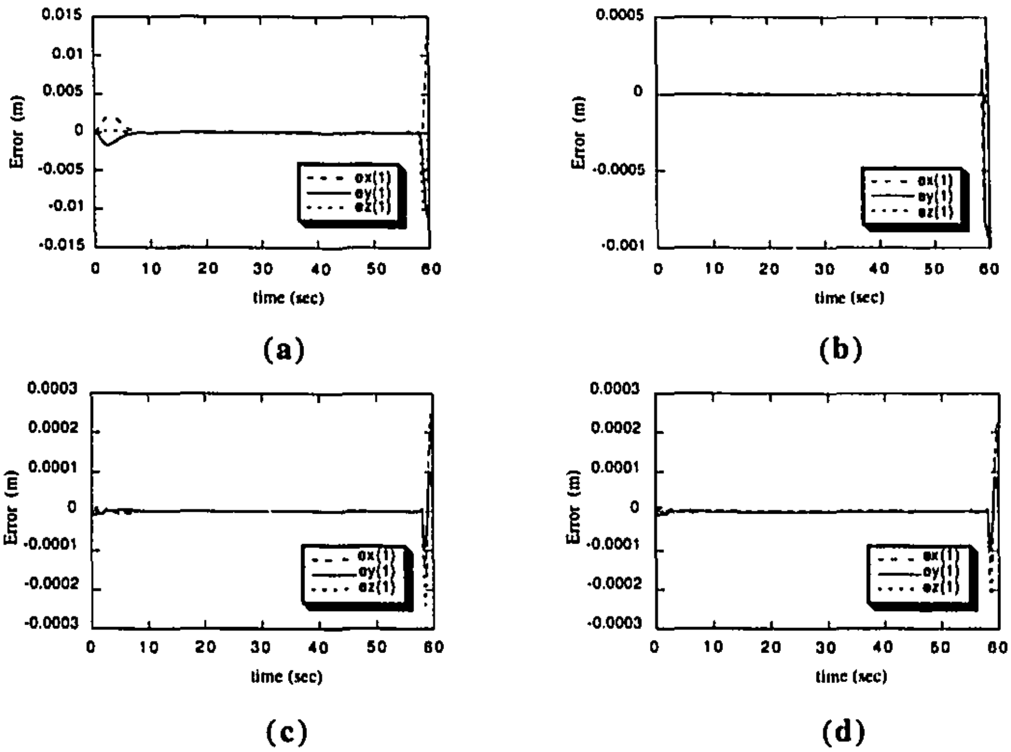


Figure 5.6: Tracking position errors for the first end-effector, (a) TJ, (b) MB, (c) MTJ1, (d) MTJ2.

Unlike the MB algorithm, the MTJ does not require any priori knowledge about system dynamics, and so it is not affected by inaccuracies in mass parameters. This becomes important when the object enters the manipulator fixed-base workspace, and the manipulators start moving ($t \approx 58$ sec). Tracking errors, which appear due to the dynamic coupling and also due to the transition phase from joint-space to task-space control, are almost five times higher for the MB, compared to those of the MTJ algorithms, see Figure 5.6. This is due to the fact that the mass properties of the control model are perturbed with respect to the true parameters by up to 5%. Note that, given enough time, tracking errors decrease and eventually vanish in all four algorithms.

Figure 5.7 shows the applied control forces/torques on the spacecraft. Comparison of the spacecraft thruster forces, shows that the peak of the required forces is about the same for the TJ and MB algorithms, while in the case of MTJ1 and MTJ2, it reaches the actuator saturation limits. The profiles of thruster forces, in most parts of the maneuver, is staircase for the MB while for the TJ algorithm, it is a smooth approximation of those profiles. For the MTJ algorithms, the profile is similar to the one of the TJ, at the beginning, and to that of the MB, at the end. This means that the value of the regulating factor which corresponds to the position error of spacecraft center of mass, is close to zero at the beginning, and almost equal to one at the end. Near the 45-th sec of the maneuver (labeled as "end of rotation maneuver" in Figure 5.7 (a)), the final desired spacecraft orientation is reached, and dynamic coupling results in small thruster forces.

As shown in Figure 5.7, in all algorithms the applied torques on the spacecraft, result in reaching actuator saturation limits of the first torque component, in attempting to compensate for the disturbances caused by manipulator motions (starting at $t \approx 58\,\mathrm{sec}$). Note that the variation of the applied torques for the MTJ algorithm is faster. Also, comparing MTJ1 to MTJ2, it can be seen that the latter results in a slightly smoother profile, which is due to more awareness of the system dynamics. However, the difference is so negligible that one may hardly decide to use MTJ2 (rather than MTJ1), considering its difficult implementation in practice as discussed before.

Figure 5.8 displays the joint torques for the first manipulator, near the end of the maneuver ($53 < t < 60$ sec). As shown in the figure, the applied torques are approximately the same for the MB and MTJ algorithms, while about 20-60% lower for the TJ algorithm.

5.4 Summary and Conclusions

This chapter presented the new Modified Transpose Jacobian (MTJ) control which, using stored data of the control command in the previous time step, yields a better performance (in

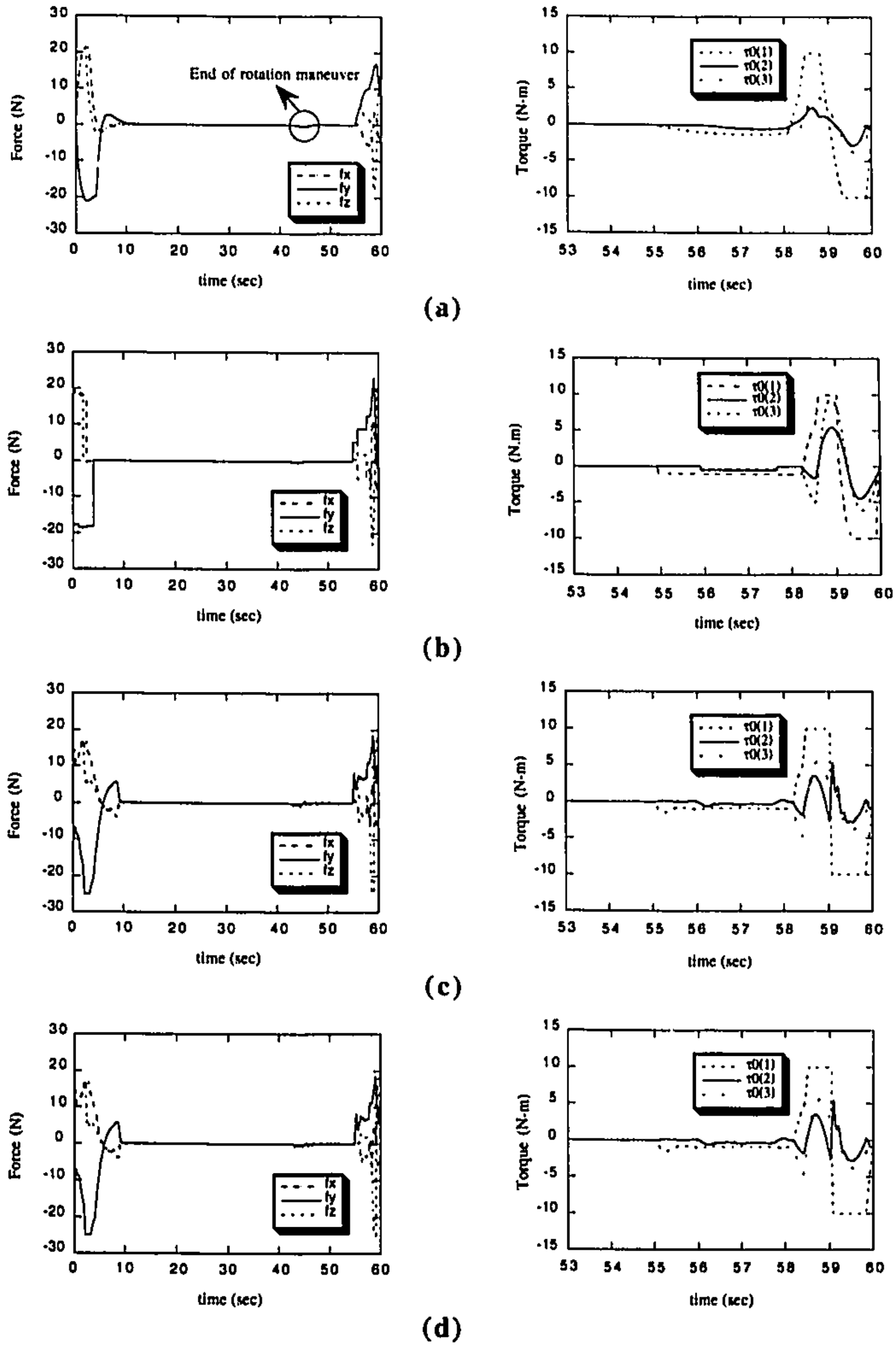


Figure 5.7: Thruster forces (left) and applied torques on the spacecraft (right), (a) TJ, (b) MB, (c) MTJ1, (d) MTJ2.

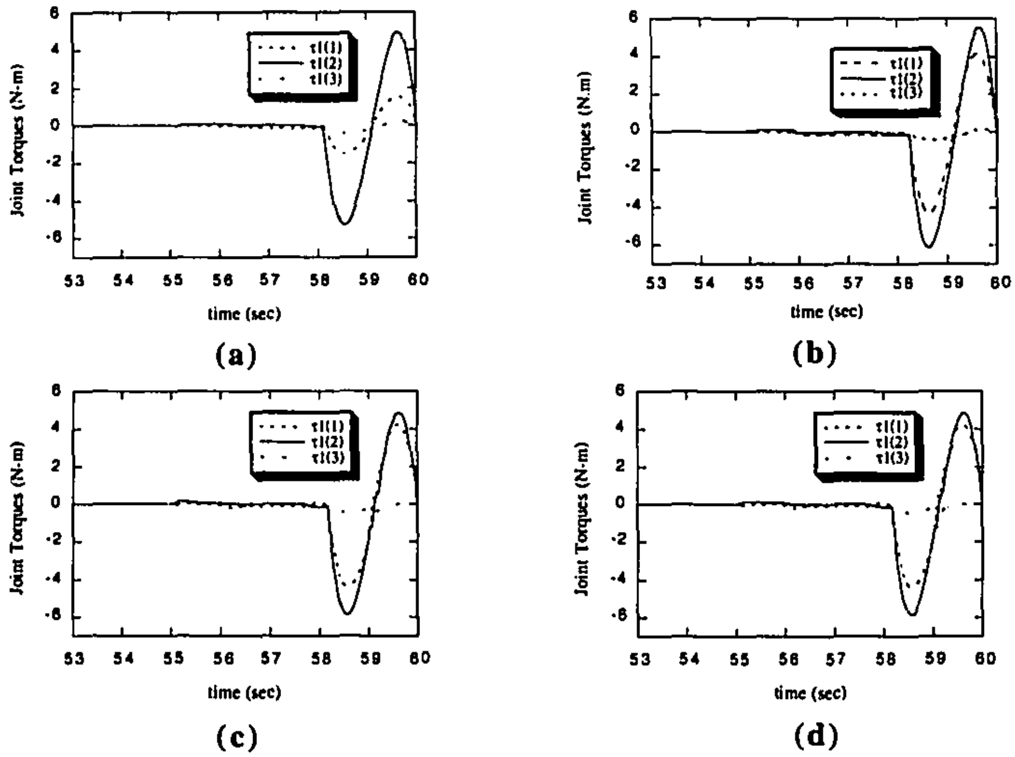


Figure 5.8: Joint torques for the first manipulator, (a) TJ, (b) MB, (c) MTJ1, (d) MTJ2.

terms of tracking errors, with the same requirements of actuator forces/torques) compared to the standard Transpose Jacobian (TJ) algorithm. The MTJ controller approximates a feedback linearization solution, with no need to a priori knowledge of the plant dynamics. Therefore, unlike a model-based algorithm, it is not affected by inaccuracies in mass properties. It was shown by simulation that the performance of the MTJ controller is comparable to that of a perfect Model-Based algorithm, with the advantage that less computational power is needed.

Unlike the standard TJ, the new MTJ algorithm works well in high speed tracking tasks. Based on presented analysis, the controller gains can be selected in a more systematic manner, and the use of high gains is avoided. In the presence of noise, it was shown by simulation that the performance of the MTJ controller is also comparable to that of a perfect MB algorithm. The substantially reduced computational requirements compared to the MB, and the good tracking and noise rejection performance characteristics in comparison with the TJ, suggest that the MTJ algorithm is a promising alternative. In particular, in those applications where model-based controllers can not be used due to computational limitations or modelling inaccuracies and uncertainties, the MTJ algorithm can be employed, with an overall performance close to that of a perfect model-based controller.

In the next chapter, manipulation of an acquired object, which can be passive or may include some internal angular momentum sources, is discussed. To this end, a new control algorithm is developed to move the captured object in accordance with a pre-determined plan which may include impacts due to contact with the environment.

Chapter 6

Multiple Impedance Control and its Application to Space Robotics

6.1 Introduction

Impedance Control was originally formulated to control dynamic interaction between a manipulator and the environment. Employing impedance control, both free motions and contact tasks can be performed without switching to different control modes. When multiple manipulators participate in a cooperative task this strategy has been formulated at the level of manipulated object, *Object Impedance Control* (OIC), to enforce a controlled impedance not of an individual arm end-point, but of the manipulated object itself. Here, a new algorithm named as *Multiple Impedance Control* (MIC) is developed, which enforces a controlled impedance of both manipulator end-points, and of a manipulated object. Physically speaking, this means that both manipulator end-effectors and the object are controlled to behave like a desired impedance in reaction to any disturbing external force on the object, and an accordant motion of different parts of the system is achieved. To manipulate space objects, the new MIC algorithm can be applied so that all participating manipulators, the free-flyer base, and the manipulated object exhibit the same impedance behavior.

First, a conceptual comparative analysis, between different control strategies, is presented. In Section 6.3, the general formulation for the MIC algorithm is derived, and based on that the tracking errors are analyzed. In addition, an estimation procedure is given for contact force determination. Then, a simple model of a robotic arm manipulating an object is considered in a thorough comparative analysis between the MIC and OIC. Then, the general MIC formulation is applied to perform a cooperative manipulation task with two fixed-base planar manipulators. The simulation results are discussed in each of these cases.

Application of the new MIC algorithm in space robotic systems is formulated in Section 6.4. As discussed before, unlike fixed-based manipulators, the base body of space robots is dynamically coupled to the arms motion. Hence, in order to control such a system, it is essential to consider this coupling between the arms and the base. For the manipulated object, inclusion of an internal source of angular momentum, is admitted. It is shown that by applying the new MIC algorithm, all participating end-effectors, the free-flyer spacecraft, and the manipulated object exhibit a similar impedance behavior. Some concluding remarks, in Section 6.5, end this chapter.

6.2 Basic Concepts

6.2.1 Problem Statement and Task Definition

Using a simple spring, Figure 6.1, the differences between various control strategies (i.e. Position, Force, and Impedance Control) are first discussed. Imposing a force F_1 at the free end of spring, A, will determine a displacement x_1 upon the value of k, and vice versa (i.e. imposing a displacement x_1 at A will determine the required force F_1). As this simple example reveals, in a mechanical system it is impossible to control both force and position along the same direction. However, using closed-loop control, we could artificially impose a desired behavior on any physical system. In other words, a desired relationship between force and motion at specific point(s) of a system can be enforced, and this is the aim of Impedance Control Laws. In our spring example, this can be achieved by setting the spring constant k.

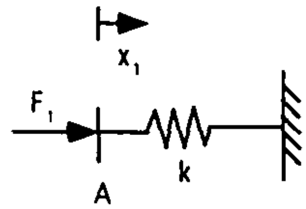


Figure 6.1: A simple spring, to visualize notion of impedance behavior.

Some definitions that are used in what follows are given here. A *manipulation task* can be defined as moving an object according to predefined trajectories which may pass through an obstacle. To compare alternative control strategies in a manipulation task, let's consider the problem in a simple form. Figure 6.2 depicts a simplified model of performing a manipulation task by a single manipulator. In case of *cooperative* operation, this simplified model can be completed by introducing a cooperation strategy to the control algorithm, and incorporating multiple manipulators.

6.2.2 Application of Alternative Control Strategies

Considering Figure 6.2, the task is defined as moving the object m_3 according to a given trajectory, $x_{3_{des}}$, by applying an appropriate force F_1 without damaging any part of the system. The manipulator is represented by m_1, connected through some spring-damper to m_2, which represents the end-effector. In this section, a *conceptual comparative analysis* between alternative control strategies is presented. To this end, the use of alternative control strategies in performing the defined task is briefly described and discussed.

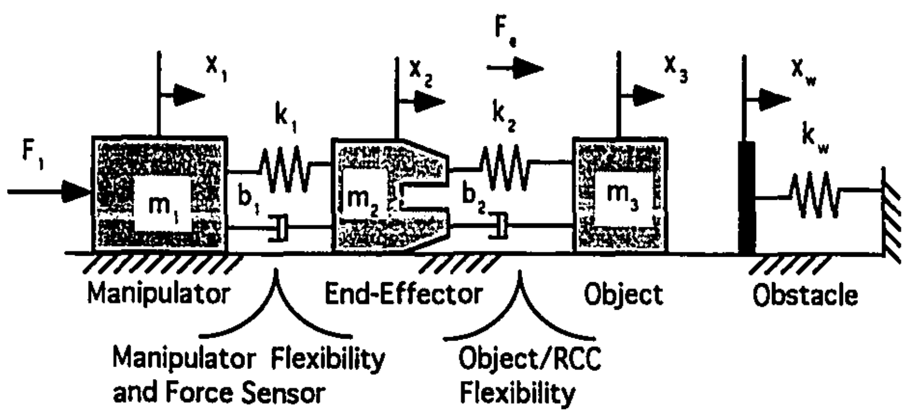


Figure 6.2: A model of performing a manipulation task by a single robot.

❐ **Position Control**, where the goal is to obtain a good tracking of either the end-effector position, x_2, (to achieve a good tracking of the object position x_3) or the object position itself, x_3. Since there is no awareness of contact between the object and an obstacle, applied forces may cause serious damage to some parts of the system.

❐ **Force (Regulation) Control**, was originally developed for performing those tasks which require direct interaction between the end-effector and its environment by regulating the end-effector force, e.g. cleaning a window. However, it may be used in object manipulation tasks by computing and applying a proper end-effector force F_e. This force is computed based on the desired object trajectory, known mass properties, and under the assumption that the object is rigid. Nevertheless, since x_2 is not regulated when controlling F_e, some tracking errors in x_3 are expected. To analyze this point, note that

$$F_e = b_2(\dot{x}_2 - \dot{x}_3) + k_2(x_2 - x_3) \tag{6.1}$$

where b_2 and k_2 are the object damping and stiffness coefficients, respectively. In case of using a Remote Centre Compliance (RCC), Craig (1989), these coefficients reflect both the object and RCC flexibility. Note that assuming negligible inertia forces for the end-effector, F_e is equal to the measured force at wrist. According to Eq. (6.1), controlling the end-effector force F_e does not yield good tracking of x_3, since F_e is also a function of x_2. For further investigation, the error in end-effector force is next computed in terms of system variables. It is shown that having this error converge to zero does not necessarily result in zero tracking error for the object position which is the original goal. To this end, the equation of motion for the object can be written as

$$m_3\ddot{x}_3 = F_o(x_3, \dot{x}_3) + F_e + F_c \tag{6.2}$$

where m_3 is the object mass, F_o includes all potential, frictional, and similar effects, and F_c is the external (contact) force applied on the object. Then, as described before, the desired end-effector force can be computed as

$$F_{e_{des}} = m_3\ddot{x}_{3_{des}} - F_o(x_{3_{des}}, \dot{x}_{3_{des}}) \tag{6.3}$$

Therefore, the corresponding error in F_e, e_f, is obtained as

$$e_f = m_3(\ddot{x}_{3_{des}} - \ddot{x}_3) - F_o(x_{3_{des}}, \dot{x}_{3_{des}}) + F_o(x_3, \dot{x}_3) + F_c \tag{6.4}$$

A well-designed force controller can make e_f go to zero. However, as Eq. (6.4) reveals, this does not necessarily yield zero tracking error for the object position, e_3

$$e_f = 0 \overset{?}{\Rightarrow} e_3 = x_{3_{des}} - x_3 = 0 \tag{6.5}$$

In free motion where no contact with the obstacle occurs, i.e. $F_c = 0$, if $e_f = 0$ it can be concluded that $\ddot{x}_3$ is close to $\ddot{x}_{3_{des}}$ under the assumption of $F_o(x_{3_{des}}, \dot{x}_{3_{des}}) = F_o(x_3, \dot{x}_3)$, see Eq. (6.4). Even so, any small deviation in acceleration will result in an integrated tracking error with time. At the time of hitting an obstacle, the contact force F_c appears with a sharp jump from zero, and a sudden change occurs in e_f which demands applying large actuator forces. If this does not result in any damage, a stable force controller results in a stop of the object at the obstacle, shortly thereafter.

❒ **Standard Impedance Control**, although formulated for performing tasks which require direct interaction between the end-effector and its environment, still it can be applied for object manipulation tasks. In so doing, enforcing a relationship between x_2 (or $\dot{x}_2$) and F_c is aimed, though the objective is good tracking of x_3. However, implementing impedance law at this level does not provide compensation for the object's inertia forces. This yields unacceptable results when the object is massive or it experiences large accelerations. It should be noted that for the standard impedance control, there is no provision for computation of the external (contact) forces applied on the object, F_c. Instead, the measured force at the wrist (which is equal to F_e, under the assumption of negligible inertia forces for the end-effector) is adapted in the impedance law. However, considering the object motion, Eq. (6.2) yields

$$F_e = m_3\ddot{x}_3 - F_o(x_3, \dot{x}_3) - F_c \tag{6.6}$$

which shows the difference between the measured force, F_e, and the real contact force, F_c. Therefore, implementing impedance law at the manipulator level ignores the possibly significant inertial effects of the object. Furthermore, even for a negligible object inertia, a relationship between x_2 and F_e is enforced (with no feedback from the object motion) which according to the previous discussion, this does not result in a good tracking of x_3.

❒ **Object Impedance Control (OIC)** is a well-formulated version of the Standard Impedance Control for object manipulation tasks. In this strategy, an impedance relationship at the object level, x_3, is enforced through feed-forward manipulator control. The novel idea here is inclusion of object inertia effects in the Impedance Control strategy. However, formulating the impedance law at the object level, with no feedback of end-effector's motion, does not yield a good tracking for flexible objects, for the same reason discussed earlier in force control and the standard impedance law. The more flexible the object is, the worse the performance of OIC will be[10].

Next, the new Multiple Impedance Control law is described and derived.

6.3 Multiple Impedance Control Law

As mentioned earlier, the basic idea in impedance control is to enforce a relationship between force and motion (position, velocity, etc.) at specific point(s) of the system. The strategy in Multiple Impedance Control, MIC, introduced for the first time in this chapter, is to enforce an equivalent impedance relationship at the manipulator end-effector level, *and* at the manipulated object level. Therefore, an object inertia effects are compensated for in the impedance law, and at the same time, the end-effector(s) tracking errors are controlled. Physically speaking, this means both the manipulator end-effector(s) and the manipulated object are controlled to respond as a designated impedance in reaction to any disturbing

10- As mentioned in Section 1.2.1, Meer and Rock (1995) have tried to solve this problem by managing different parameters in implementing the algorithm.

external force on the object, and different parts of the system are led to an accordant motion. For mobile manipulators, e.g. space free-flyers, the new MIC algorithm is applied so that all participating manipulators, the moving platform (base), and the manipulated object exhibit the same impedance behavior, as implied by "multiple" in the MIC.

While OIC enforces an impedance law on the object motion, MIC enforces an impedance law on both the manipulator end-effector(s), and the manipulated object. This major difference between the MIC and OIC allows for proper trajectory planning of the end-effector(s), based on the desired trajectory for the object and the grasp condition. Note that for the case of a redundant system, the end-effector(s) trajectory can be planned so as to optimize the performance. Other differences between the MIC and OIC include allowing for a difference between the *contact force* and other external forces which are applied on the object, as well as improved *contact force estimation*.

In this section, the general formulation of the new MIC algorithm is derived for fixed-base cooperative manipulators. An estimation procedure for the contact force determination is discussed, and tracking errors are analyzed. Considering a simple model for manipulating an object with a single robotic arm, as discussed in the previous section, a comparative analysis between the MIC and OIC is presented. Root locus analyses, and simulation results are given in each case. Then, the application of the MIC algorithm to perform a cooperative manipulation task with two fixed-base planar manipulators is discussed, and simulated.

6.3.1 General Formulation

Performing a cooperative manipulation task, as defined in the previous section, requires coordination between participating robotic arms, Figure 6.3. To this end, the dynamics equations of each participating manipulator can be written as

$$\mathbf{H}^{(i)}(\mathbf{q}^{(i)})\ddot{\mathbf{q}}^{(i)}+\mathbf{C}^{(i)}(\mathbf{q}^{(i)},\dot{\mathbf{q}}^{(i)})=\mathbf{Q}^{(i)} \tag{6.7}$$

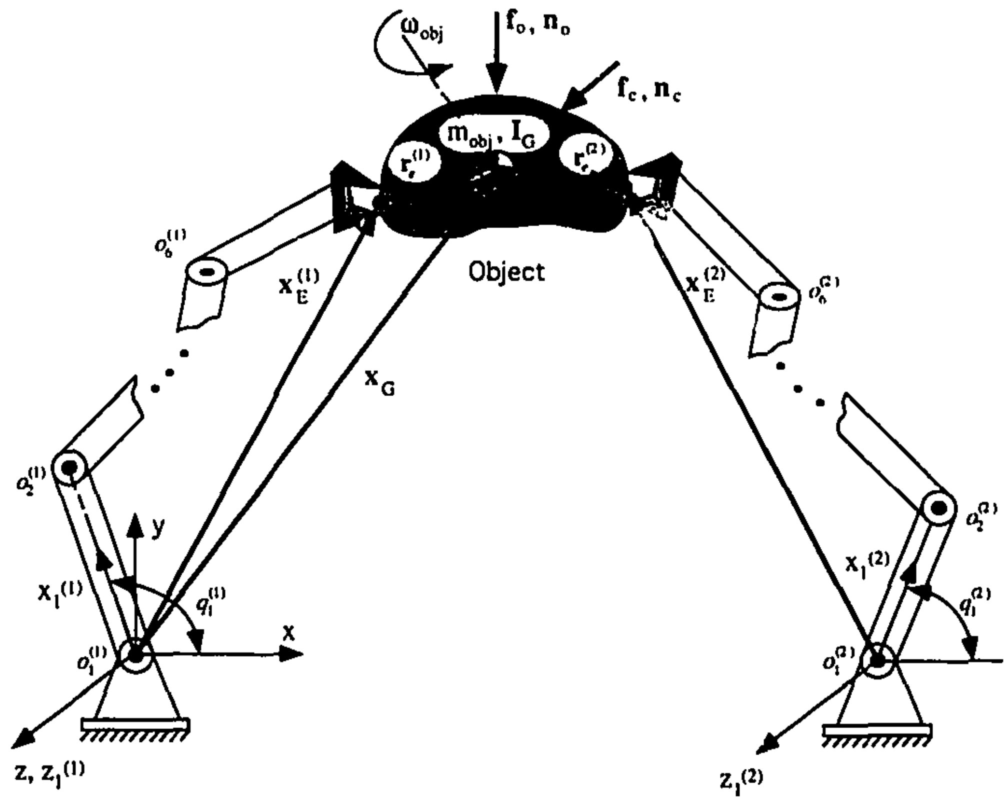


Figure 6.3: Two robotic arms performing a cooperative manipulation task.

where the superscript "i" corresponds to the i-th manipulator, and the vector of joint angles and displacements is chosen as generalized coordinates $\mathbf{q}^{(i)}$. Note that $\mathbf{C}^{(i)}$ contains all the gravity and nonlinear velocity terms, where in a microgravity environment the gravity terms are practically zero. Assuming that each manipulator has six DOF, and using a square Jacobian $\mathbf{J}_C^{(i)}$, the output speeds ($\dot{\tilde{\mathbf{x}}}$) are computed in terms of the generalized ones ($\dot{\mathbf{q}}$) as

$$\dot{\tilde{\mathbf{x}}}^{(i)} = \mathbf{J}_C^{(i)} \dot{\mathbf{q}}^{(i)} \tag{6.8a}$$

where

$$\tilde{\mathbf{x}}^{(i)} = \begin{Bmatrix} \mathbf{x}_E^{(i)} \\ \boldsymbol{\delta}_E^{(i)} \end{Bmatrix} \tag{6.8b}$$

$\mathbf{x}_E^{(i)}$ describes the i-th end-effector position, and $\boldsymbol{\delta}_E^{(i)}$ is a set of Euler angles which describes the i-th end-effector orientation. The equations of motion, Eq. (6.7), can then be written in terms of the output coordinates $\tilde{\mathbf{x}}^{(i)}$, as

$$\tilde{\mathbf{H}}^{(i)}(\mathbf{q}^{(i)})\ddot{\mathbf{x}}^{(i)}+\tilde{\mathbf{C}}^{(i)}(\mathbf{q}^{(i)},\dot{\mathbf{q}}^{(i)})=\tilde{\mathbf{Q}}^{(i)} \tag{6.9a}$$

where

$$\tilde{\mathbf{H}}^{(i)}=\mathbf{J}_C^{(i)-T}\,\mathbf{H}^{(i)}\,\mathbf{J}_C^{(i)-1} \qquad \tilde{\mathbf{C}}^{(i)}=\mathbf{J}_C^{(i)-T}\mathbf{C}^{(i)}-\tilde{\mathbf{H}}^{(i)}\,\dot{\mathbf{J}}_C^{(i)}\,\dot{\mathbf{q}}^{(i)} \qquad \tilde{\mathbf{Q}}^{(i)}=\mathbf{J}_C^{(i)-T}\mathbf{Q}^{(i)} \tag{6.9b}$$

The vector of generalized forces in the task space, $\tilde{\mathbf{Q}}^{(i)}$, can be written as

$$\tilde{\mathbf{Q}}^{(i)}=\tilde{\mathbf{Q}}_{app}^{(i)}+\tilde{\mathbf{Q}}_{react}^{(i)} \tag{6.10a}$$

where $\tilde{\mathbf{Q}}_{react}^{(i)}$ is the reaction load on the end-effector, and $\tilde{\mathbf{Q}}_{app}^{(i)}$ is the applied controlling force which is divided into two parts, *motion-concerned* and *force-concerned* as

$$\tilde{\mathbf{Q}}_{app}^{(i)}=\tilde{\mathbf{Q}}_m^{(i)}+\tilde{\mathbf{Q}}_f^{(i)} \tag{6.10b}$$

where $\tilde{\mathbf{Q}}_m^{(i)}$ is the applied control force concerning the motion of the end-effector, while $\tilde{\mathbf{Q}}_f^{(i)}$ is the required force to be applied on the manipulated object by the end-effector. To obtain proper expressions for these terms, let's first consider the equations of motion for the manipulated object.

The equations of motion for the object can be written as

$$\begin{aligned} m_{obj}\,\ddot{\mathbf{x}}_G &= \mathbf{f}_c+\mathbf{f}_o+\sum_{i=1}^{n}\mathbf{f}_e^{(i)} \\ \mathbf{I}_G\dot{\boldsymbol{\omega}}_{obj}+\boldsymbol{\omega}_{obj}\times\mathbf{I}_G\,\boldsymbol{\omega}_{obj} &= \mathbf{n}_c+\mathbf{n}_o+\sum_{i=1}^{n}\mathbf{r}_e^{(i)}\times\mathbf{f}_e^{(i)}+\sum_{i=1}^{n}\mathbf{n}_e^{(i)} \end{aligned} \tag{6.11}$$

where m_{obj} is the object mass, n is the number of participating manipulators in the manipulation task, $\mathbf{I}_G$ is its moment of inertia about center of mass, $\ddot{\mathbf{x}}_G$ is acceleration of center of mass, $\boldsymbol{\omega}_{obj}$ is the object angular velocity, $\dot{\boldsymbol{\omega}}_{obj}$ is the object angular acceleration, $\mathbf{f}_c$ is the force applied on the object due to contact with the environment, $\mathbf{f}_e^{(i)}$ is the i-th end-effector force exerted on the object, $\mathbf{f}_o$ is the vector of other external forces applied on the object (including gravity forces), $\mathbf{n}_c$ is the contact torque applied on the object about its center of mass (including the moment of $\mathbf{f}_c$), $\mathbf{r}_e^{(i)}$ is the position vector of the i-th end-effector with respect to the object center of mass, $\mathbf{n}_e^{(i)}$ is the i-th end-effector torque exerted on the object about its center of mass, and $\mathbf{n}_o$ is the vector of other external torque applied on the object (including the moment of $\mathbf{f}_o$) about its center of mass. Similar to Eq. (3.7), choosing a set of Euler angles that describes the orientation of the object, $\boldsymbol{\delta}_{obj}$, the object angular velocity can be expressed in terms of Euler rates as

$$\boldsymbol{\omega}_{obj} = \mathbf{S}_{obj}\,\dot{\boldsymbol{\delta}}_{obj} \tag{6.12}$$

which can be substituted into Eq. (6.11), to obtain the equations of motion for the object as

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}_\omega = \mathbf{F}_c + \mathbf{F}_o + \mathbf{G}\,\mathbf{F}_e \tag{6.13a}$$

where

$$\ddot{\mathbf{x}} = \begin{Bmatrix} \ddot{\mathbf{x}}_G \\ \ddot{\boldsymbol{\delta}}_{obj} \end{Bmatrix} \quad \mathbf{M} = \begin{bmatrix} m_{obj}\mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{S}_{obj}^T\,\mathbf{I}_G\,\mathbf{S}_{obj} \end{bmatrix} \quad \mathbf{F}_\omega = \begin{Bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{S}_{obj}^T\left(\left[\boldsymbol{\omega}_{obj}\right]^\times \mathbf{I}_G\,\boldsymbol{\omega}_{obj} + \mathbf{I}_G\,\dot{\mathbf{S}}_{obj}\,\dot{\boldsymbol{\delta}}_{obj}\right) \end{Bmatrix}$$

$$\mathbf{F}_c = \begin{Bmatrix} \mathbf{f}_c \\ \mathbf{S}_{obj}^T\,\mathbf{n}_c \end{Bmatrix} \quad \mathbf{F}_o = \begin{Bmatrix} \mathbf{f}_o \\ \mathbf{S}_{obj}^T\,\mathbf{n}_o \end{Bmatrix} \quad \mathbf{F}_e = \begin{Bmatrix} \mathbf{F}_e^{(1)} \\ \vdots \\ \mathbf{F}_e^{(n)} \end{Bmatrix}_{6n\times1} \quad \mathbf{F}_e^{(i)} = \begin{Bmatrix} \mathbf{f}_e^{(i)} \\ \mathbf{n}_e^{(i)} \end{Bmatrix}_{6\times1} \tag{6.13b}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{S}_{obj}^T[\mathbf{r}_e^{(1)}]^\times_{3\times3} & \mathbf{S}_{obj}^T & & \mathbf{S}_{obj}^T[\mathbf{r}_e^{(n)}]^\times_{3\times3} & \mathbf{S}_{obj}^T \end{bmatrix}_{6\times6n}$$

The matrix **M** will be referred to as the mass matrix, and the matrix **G** as the grasp matrix.

A desired impedance relationship for the object motion is chosen as

$$\mathbf{M}_{des}\ddot{\mathbf{e}} + \mathbf{k}_d\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} = -\mathbf{F}_c \tag{6.14}$$

where $\mathbf{M}_{des}$ is the object desired mass matrix, $\mathbf{e} = (\mathbf{x}_{des} - \mathbf{x})$ is the object position/orientation error vector, and $\mathbf{k}_p$ and $\mathbf{k}_d$ are gain matrices (which are usually selected as diagonal matrices). Comparing Eq. (6.14) to Eq. (6.13), it can be seen that the desired impedance behavior can be obtained if

$$\mathbf{G}\mathbf{F}_{e_{req}} = \mathbf{M}\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des} + \mathbf{k}_d\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \mathbf{F}_c\right) + \mathbf{F}_\omega - \left(\mathbf{F}_c + \mathbf{F}_o\right) \tag{6.15}$$

provided that $\mathbf{S}_{obj}$ is not singular which is a matter of Euler angles definition. In other words, applying the required end-effector forces/torques, $\mathbf{F}_{e_{req}}$, on the object results in the targeted impedance relationship as described in Eq. (6.14). Now, Eq. (6.15) can be solved to obtain a *minimum norm solution*. Therefore, the required end-effector forces are obtained as

$$\mathbf{F}_{e_{req}} = \mathbf{G}^{\#}\left\{\mathbf{M}\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des} + \mathbf{k}_d\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \mathbf{F}_c\right) + \mathbf{F}_{\omega} - \left(\mathbf{F}_c + \mathbf{F}_o\right)\right\} \tag{6.16a}$$

where $\mathbf{G}^{\#}$ is the pseudoinverse of the grasp matrix $\mathbf{G}$, a full-rank matrix (provided that $\mathbf{S}_{obj}$ is not singular), defined as

$$\mathbf{G}^{\#} = \mathbf{W}^{-1}\mathbf{G}^{T}\left(\mathbf{G}\mathbf{W}^{-1}\mathbf{G}^{T}\right)^{-1} \tag{6.16b}$$

weighted by a task weighting matrix $\mathbf{W}$, so that linear and angular motions or their components can have different weights. Assuming that $\mathbf{F}_o$ and the object mass and geometric properties are known, computation of $\mathbf{F}_{e_{req}}$ requires knowing the value of the contact force, $\mathbf{F}_c$. Since, in general it is not possible to measure this force, it has to be estimated, see Section 6.3.3. Therefore, Eq. (6.16a) can be written as

$$\mathbf{F}_{e_{req}} = \mathbf{G}^{\#}\left\{\mathbf{M}\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des} + \mathbf{k}_d\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \hat{\mathbf{F}}_c\right) + \mathbf{F}_{\omega} - \left(\hat{\mathbf{F}}_c + \mathbf{F}_o\right)\right\} \tag{6.17}$$

where $\hat{\mathbf{F}}_c$ is the estimated value of the contact force $\mathbf{F}_c$. Note that based on the grasp condition, it may be required to apply additional internal forces and moments on the object, $\mathbf{F}_{int}$. Then, Eq. (6.16) can be modified to

$$\mathbf{F}_{e_{req}} = \mathbf{G}^{\#}\left\{\mathbf{M}\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des} + \mathbf{k}_d\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \hat{\mathbf{F}}_c\right) + \mathbf{F}_{\omega} - \left(\hat{\mathbf{F}}_c + \mathbf{F}_o\right)\right\} + \left(\mathbf{1} - \mathbf{G}^{\#}\mathbf{G}\right)\mathbf{F}_{int} \tag{6.18}$$

where $\mathbf{1}$ is a 6n×6n identity matrix. Note that $\mathbf{F}_{int}$ does not affect the object motion, since the added term is in the null space of the grasp matrix $\mathbf{G}$.

Now, based on the definition of $\mathbf{F}_e$, Eq. (6.13b), the force which has to be supplied by the i-th end-effector, $\mathbf{F}_{e_{req}}^{(i)}$, is directly obtained from $\mathbf{F}_{e_{req}}$. This yields the force-concerned part of the applied controlling force, according to the definition given in Eq. (6.10), as

$$\tilde{\mathbf{Q}}_f^{(i)} = \mathbf{F}_{e_{req}}^{(i)} \tag{6.19}$$

Note that $\tilde{\mathbf{Q}}_f^{(i)}$ is virtually canceled by the reaction load on each end-effector. On the other hand, the reaction load is obtained as

$$\tilde{\mathbf{Q}}_{react}^{(i)} = -\mathbf{F}_e^{(i)} \tag{6.20a}$$

where

$$\mathbf{F}_e = \mathbf{G}^{\#}\left[\mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}_{\omega} - (\mathbf{F}_c + \mathbf{F}_o)\right] + \left(\mathbf{1} - \mathbf{G}^{\#}\mathbf{G}\right)\mathbf{F}_{int} \tag{6.20b}$$

Next, we have to obtain a proper expression for the motion-concerned part of the applied controlling force $\tilde{\mathbf{Q}}_m^{(i)}$.

As discussed earlier, in the MIC algorithm the same impedance law is imposed on the behavior of both the end-effector(s) and the manipulated object. Therefore, similar to Eq. (6.14), the impedance law for the i-th end-effector can be written as

$$\mathbf{M}_{des}\ddot{\tilde{\mathbf{e}}}^{(i)} + \mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)} + \mathbf{k}_p\tilde{\mathbf{e}}^{(i)} = -\mathbf{F}_c \tag{6.21}$$

where $\tilde{\mathbf{e}}^{(i)} = \tilde{\mathbf{x}}_{des}^{(i)} - \tilde{\mathbf{x}}^{(i)}$ is the i-th end-effector position/orientation error vector, and the rest has been already defined. Then, $\tilde{\mathbf{Q}}_m^{(i)}$ can be obtained similar to the above derivation of $\tilde{\mathbf{Q}}_f^{(i)}$, as

$$\tilde{\mathbf{Q}}_m^{(i)} = \tilde{\mathbf{H}}^{(i)}(\mathbf{q}^{(i)})\mathbf{M}_{des}^{-1}\left[\mathbf{M}_{des}\ddot{\tilde{\mathbf{x}}}_{des}^{(i)} + \mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)} + \mathbf{k}_p\tilde{\mathbf{e}}^{(i)} + \mathbf{F}_c\right] + \tilde{\mathbf{C}}^{(i)}(\mathbf{q}^{(i)},\dot{\mathbf{q}}^{(i)}) \tag{6.22a}$$

or (substituting the estimated value for the contact force)

$$\tilde{\mathbf{Q}}_m^{(i)} = \tilde{\mathbf{H}}^{(i)}(\mathbf{q}^{(i)})\mathbf{M}_{des}^{-1}\left[\mathbf{M}_{des}\ddot{\tilde{\mathbf{x}}}_{des}^{(i)} + \mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)} + \mathbf{k}_p\tilde{\mathbf{e}}^{(i)} + \hat{\mathbf{F}}_c\right] + \tilde{\mathbf{C}}^{(i)}(\mathbf{q}^{(i)},\dot{\mathbf{q}}^{(i)}) \tag{6.22b}$$

Note that the desired trajectory for the i-th end-effector motion, $\tilde{\mathbf{x}}_{des}^{(i)}$, can be defined based on the desired trajectory for the object motion, the object geometry, and the grasp condition. In other words, based on the *grasp constraints* defined as

$$\mathbf{g}^{(i)}(\mathbf{x}_{des}, \tilde{\mathbf{x}}_{des}^{(i)}) = \mathbf{0} \qquad i = 1, \cdots, n \tag{6.23}$$

and the object desired trajectory, $\mathbf{x}_{des}$, the desired end-effectors trajectories can be determined. Substituting Eqs. (6.22), and (6.19) into Eq. (6.10b), the applied controlling force can be computed. The block diagram of the MIC algorithm is demonstrated in Figure 6.4.

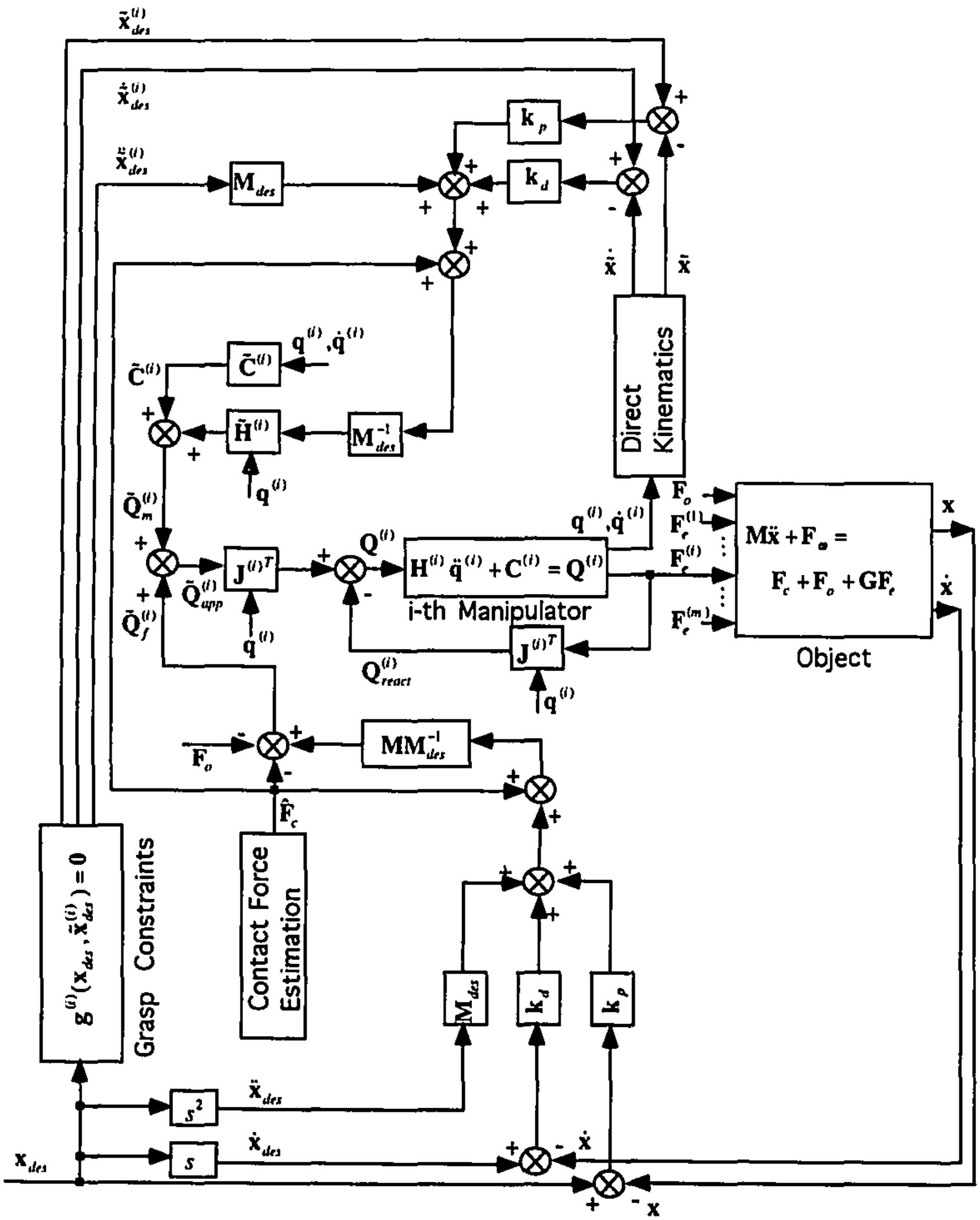


Figure 6.4: The block diagram of the MIC algorithm.

6.3.2 Error Analysis

Substituting Eqs. (6.22), (6.20), and (6.19) into Eq. (6.10), and then the result into Eq. (6.9), yields

$$\begin{aligned}&\tilde{\mathbf{H}}^{(i)}(\mathbf{q}^{(i)})\left\{\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\tilde{\mathbf{x}}}_{des}^{(i)}+\mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)}+\mathbf{k}_p\tilde{\mathbf{e}}^{(i)}+\hat{\mathbf{F}}_c\right)-\ddot{\tilde{\mathbf{x}}}^{(i)}\right\}+\\ &\qquad\mathbf{G}^{\#}\mathbf{M}\left\{\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\hat{\mathbf{F}}_c\right)-\ddot{\mathbf{x}}\right\}=\mathbf{0}\end{aligned} \tag{6.24a}$$

where it is assumed that mass and geometric properties for the manipulated object and manipulator are known. Also, it is assumed that the contact force estimation procedure yields an exact value for this force. Since Eq. (6.24a) must hold for any $\mathbf{M}$ and $\tilde{\mathbf{H}}^{(i)}$, it can be concluded that

$$\begin{aligned}&\tilde{\mathbf{H}}^{(i)}(\mathbf{q}^{(i)})\left\{\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\tilde{\mathbf{x}}}_{des}^{(i)}+\mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)}+\mathbf{k}_p\tilde{\mathbf{e}}^{(i)}+\hat{\mathbf{F}}_c\right)-\ddot{\tilde{\mathbf{x}}}^{(i)}\right\}=\mathbf{0} \qquad i=1,\cdots,n\\ &\mathbf{G}^{\#}\mathbf{M}\left\{\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\hat{\mathbf{F}}_c\right)-\ddot{\mathbf{x}}\right\}=\mathbf{0}\end{aligned} \tag{6.24b}$$

Then, since $\mathbf{G}^{\#}$ is of full-rank, this results in

$$\begin{aligned}&\tilde{\mathbf{H}}^{(i)}(\mathbf{q}^{(i)})\left\{\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\tilde{\mathbf{x}}}_{des}^{(i)}+\mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)}+\mathbf{k}_p\tilde{\mathbf{e}}^{(i)}+\hat{\mathbf{F}}_c\right)-\ddot{\tilde{\mathbf{x}}}^{(i)}\right\}=\mathbf{0} \qquad i=1,\cdots,n\\ &\mathbf{M}\left\{\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\hat{\mathbf{F}}_c\right)-\ddot{\mathbf{x}}\right\}=\mathbf{0}\end{aligned} \tag{6.25}$$

Finally, noting the fact that $\mathbf{M}$ and $\tilde{\mathbf{H}}^{(i)}$ are positive definite mass matrices, Eq. (6.25) results in

$$\begin{aligned}&\mathbf{M}_{des}\ddot{\tilde{\mathbf{e}}}^{(i)}+\mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)}+\mathbf{k}_p\tilde{\mathbf{e}}^{(i)}+\hat{\mathbf{F}}_c=\mathbf{0} \qquad i=1,\cdots,n\\ &\mathbf{M}_{des}\ddot{\mathbf{e}}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\hat{\mathbf{F}}_c=\mathbf{0}\end{aligned} \tag{6.26}$$

which means all participating manipulators and the manipulated object exhibit the same designated impedance behavior. Note that the MIC approach permits choosing different impedance parameters for the object dynamical behavior and the end-effectors (by selecting $\mathbf{M}_{des}$, $\mathbf{k}_d$, and $\mathbf{k}_p$ in Eq. (6.21) different from those of Eq. (6.14)). However, physical intuition as well as simulation analyses indicate that the best results are achieved by choosing equivalent parameters. This is due to the fact that enforcing the same pre-set impedance on different parts of the system results in *accordant* motion throughout the system while executing a manipulation task. Harmonic motion of the end-effectors and manipulated object is ensured via same error dynamics as obtained above.

6.3.3 Contact Force Estimation

As mentioned in the previous section, computation of $\mathbf{F}_{e_{req}}$ requires knowing the value of the contact force, $\mathbf{F}_c$. In general, this has to be estimated, and this is the focus of this section.

To compute the contact force, Eq. (6.13) can be rewritten as

$$\mathbf{F}_c = \mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}_\omega - \mathbf{F}_o - \mathbf{G}\mathbf{F}_e \tag{6.27}$$

It is assumed that $\mathbf{F}_o$, and also the object mass and geometric properties are known. Assuming that end-effectors are equipped with force sensors, $\mathbf{F}_e$ can be measured and substituted into this equation. Also, based on measurements of object motion, $\mathbf{F}_\omega$ can be computed as given in Eq. (6.13b), and substituted into Eq. (6.27). However, to evaluate the contact force, the object acceleration must be also known. Since this is not usually measured, it can be approximated through a numerical procedure. In OIC implementation, either the desired acceleration, or the last *commanded acceleration* which is defined as

$$\ddot{\mathbf{x}}_{cmd} = \mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des} + \mathbf{k}_d\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \hat{\mathbf{F}}_c\right) \tag{6.28}$$

are used. Schneider and Cannon (1992) describe that both of these two approximations yield acceptable experimental results, though they have emphasized that a more sophisticated procedure would improve the performance. In fact, since there may be a considerable difference between $\ddot{\mathbf{x}}$ and $\ddot{\mathbf{x}}_{des}$, particularly after contact, using $\ddot{\mathbf{x}}_{des}$ does not yield reliable results. On the other hand, using Eq. (6.28) may result in a poor approximation because of sudden variations in contact force (at each contact).

Here, the suggestion is a direct usage of finite difference approximation as

$$\ddot{\mathbf{x}} = \frac{\dot{\mathbf{x}}_t - \dot{\mathbf{x}}_{t-\Delta t}}{\Delta t} \tag{6.29a}$$

or

$$\ddot{\mathbf{x}} = \frac{\mathbf{x}_t - 2\mathbf{x}_{t-\Delta t} + \mathbf{x}_{t-2\Delta t}}{(\Delta t)^2} \tag{6.29b}$$

where Δt is the time step used in the estimation procedure. Note that because of practical reasons (i.e. time requirement for measurements and corresponding calculations), Δt can not be infinitesimally close to zero. At sufficiently high sampling rates, this does not introduce a significant error, even during contact. Substituting Eq. (6.29) for acceleration, the contact force can be estimated based on Eq. (6.27) as

$$\hat{\mathbf{F}}_c = \mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}_\omega - \mathbf{F}_o - \mathbf{G}\mathbf{F}_e \tag{6.30}$$

Next, the system depicted in Figure 6.2 is considered for a comparative analysis between the MIC and OIC algorithms.

6.3.4 Case Study: A Comparative Analysis (Single Manipulator)

The single robotic arm manipulating an object discussed in Section 6.2, is used here to compare the nature and performance of the MIC and OIC algorithms. First, the system dynamics model is derived, and then the controllability of the system is investigated. The MIC and OIC laws are implemented, and compared through root locus analyses. The system is then simulated under both control laws, and the simulation results are compared.

6.3.4.1 Dynamics Model

For the 3-DOF system depicted in Figure 6.2, the equations of motion are

$$\begin{aligned}
&m_1\ddot{x}_1 + b_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2 + l_{1_{free}}) = F_1 \\
&m_2\ddot{x}_2 + b_1(\dot{x}_2 - \dot{x}_1) + b_2(\dot{x}_2 - \dot{x}_3) + k_1(x_2 - x_1 - l_{1_{free}}) + k_2(x_2 - x_3 + l_{2_{free}}) = 0 \\
&m_3\ddot{x}_3 + b_2(\dot{x}_3 - \dot{x}_2) + k_2(x_3 - x_2 - l_{2_{free}}) = f_o + f_c
\end{aligned} \tag{6.31}$$

where $l_{1_{free}}$, and $l_{2_{free}}$ are the free lengths of springs k_1, and k_2, respectively, f_c is the contact force, and f_o is the resultant of other external forces applied on the object. Gravity effects are neglected, and all mass and stiffness properties are assumed to be known.

State-space representation of Eq. (6.31) can be written as

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\,\mathbf{x} + \mathbf{b}\,u + \mathbf{w} \\ \mathbf{y} &= \mathbf{c}\mathbf{x} + \mathbf{d}\,u\end{aligned} \tag{6.32}$$

where $\mathbf{x} = (x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3)^T$, $u = F_1$, and

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -k_1/m_1 & k_1/m_1 & 0 & -b_1/m_1 & b_1/m_1 & 0 \\ k_1/m_2 & -(k_1+k_2)/m_2 & k_2/m_2 & b_1/m_2 & -(b_1+b_2)/m_2 & b_2/m_2 \\ 0 & k_2/m_3 & -k_2/m_3 & 0 & b_2/m_3 & -b_2/m_3 \end{bmatrix} \tag{6.33a}$$

$$\mathbf{b} = \left(0, 0, 0, 1/m_1, 0, 0\right)^T \tag{6.33b}$$

$$\mathbf{w} = \left(0, 0, 0, \frac{-k_1 l_{1free}}{m_1}, \frac{(k_1 l_{1free} - k_2 l_{2free})}{m_2}, \frac{(f_o + f_c + k_2 l_{2free})}{m_3}\right)^T \tag{6.33c}$$

and the output vector, $\mathbf{y}$, for each algorithm can be chosen accordingly.

6.3.4.2 Controllability of the system

The controllability matrix of the system is defined as[11]

$$\widehat{\mathbf{C}} = \begin{bmatrix}\mathbf{b} & \mathbf{A}\mathbf{b} & \mathbf{A}^2\mathbf{b} & \mathbf{A}^3\mathbf{b} & \mathbf{A}^4\mathbf{b} & \mathbf{A}^5\mathbf{b}\end{bmatrix} \tag{6.34}$$

The determinant of $\widehat{\mathbf{C}}$ is calculated as

$$\left|\widehat{\mathbf{C}}\right| = \frac{k_1^2 k_2^2 \left(k_1^2 m_2 m_3 + (k_2 b_1^2 - k_1 b_1 b_2)(m_2 + m_3)\right)}{m_1^6 m_2^5 m_3^3} \tag{6.35}$$

In general, $\left|\widehat{\mathbf{C}}\right|$ is not zero which implies that $\widehat{\mathbf{C}}$ is a full-rank matrix. This means that the system is controllable, and that the actuator is able to take the system states to any desired

11- See Takahashi, Y. (1970).

configuration in a finite time, provided that a proper input function, $u(t)$, is selected. This observation motivates further study in controlling both the manipulator and object, using the MIC algorithm. In the following, both OIC and MIC algorithms are applied, to provide the input function, $u(t)$, and control this system.

6.3.4.3 Root locus analysis

To investigate the stability and performance of the MIC compared to OIC law, for the system depicted in Figure 6.2, a root locus analysis is presented in this section. To this end, a root locus parameter has to be selected. Then, the poles of the corresponding transfer functions, $G_{MIC}(s)$ and $G_{OIC}(s)$, i.e. roots of the characteristic equation in each case, are calculated for a set of values for the chosen parameter. Here, the object stiffness coefficient k_2 is selected as variable parameter, where $G_{MIC}(s)$ and $G_{OIC}(s)$, and the corresponding characteristic equations are presented in Appendix C.

For a rigid system, i.e. $k_1, k_2 \to \infty$, considering Eqs. (C.2-7) it is obtained

$$\begin{aligned} G(s) &= \frac{x_3}{x_{des_3}} = \lim_{k_1, k_2 \to \infty} G_{MIC}(s) = \lim_{k_1, k_2 \to \infty} G_{OIC}(s) \\ &= \frac{(m_{des}s^2 + k_d s + k_p)(\hat{m}_1 + \hat{m}_2 + \hat{m}_3)}{(m_{des}s^2 + k_w)(m_1 + m_2 + m_3) + (k_d s + k_p)(\hat{m}_1 + \hat{m}_2 + \hat{m}_3)} \end{aligned} \tag{6.36}$$

which means that for a rigid system, both algorithms yield the same closed-loop transfer function. If the given mass parameters for control purposes are the same as true ones, i.e. $\hat{m}_i = m_i$, then G(s)=1.0 in free motion ($k_w = 0$); there is a perfect tracking.

Given that the true and given mass parameters are all positive, and applying the Routh-Hurwitz criterion, all of the zeros and poles of Eq. (6.36) lie in the left half of the s-plane if and only if

$$m_{des} > 0 \quad \& \quad k_d > 0 \quad \& \quad k_p > 0 \tag{6.37}$$

and upon this condition, both algorithms are stable for a rigid system.

Note that considering Eqs. (C.4b) and (C.7b), the sum of the roots of characteristic equation (s_i) for the MIC and OIC can be written as

❐ **For the MIC:**

$$\sum_{i=1}^{6} s_i = \frac{-\left(\hat{m}_1 m_2 m_3 k_d + m_{des}(m_1 + m_2) m_3 b_1 + m_{des}(m_2 + m_3) m_1 b_2\right)}{m_{des} m_1 m_2 m_3} \tag{6.38}$$

❐ **For the OIC:**

$$\sum_{i=1}^{6} s_i = \frac{-\left((m_1 + m_2) m_3 b_1 + (m_2 + m_3) m_1 b_2\right)}{m_1 m_2 m_3} \tag{6.39}$$

As revealed by Eq. (6.39), the sum of the roots for the OIC algorithm is a function of system parameters only, and is mostly affected by the damping characteristics of the system; the controller parameters do not affect the sum of the roots. However, it is seen that for the MIC, this sum is also a function of k_d and m_{des}, and this permits easier pole adjustment.

As shown in Appendix D, the root loci for the MIC and OIC algorithms, as a function of the object stiffness (k_2) for various damping factors (b_2), reveal that for a relatively well-damped object both algorithms are stable, whether or not the object is in contact with the obstacle. However, for an object with light damping, the OIC algorithm becomes unstable if there is no contact. Note that a contact between the object and an obstacle, adds a feedback effect to the system, and so its dynamic behavior changes. Considering this unstable case, the effect of different controller parameters on the stability of the OIC algorithm is investigated in Appendix D. It is shown that choosing larger gains, solely, does not result in a stable system. However, a larger desired mass value has a positive effect on the stability of the OIC algorithm, though larger inertia of the desired object impedance results in slower performance, as will be shown by simulation. For an undamped object, i.e. $b_2 = 0$, it is shown that the MIC algorithm is stable (whether or not the object is in contact with the obstacle), while the OIC becomes unstable. In this case, choosing a larger value for the desired mass or a larger damping gain does not yield a stable system. Based on this analysis, it can be concluded that with respect to system stability, the MIC algorithm is always preferred, compared to the OIC.

Next, the performance of the two algorithms is simulated and compared, in the case where the system parameters are chosen so that stability is ensured in both *no contact* and *in contact* phases.

6.3.4.4 Simulation Results

The system depicted in Figure 6.2, is now simulated under the MIC and OIC laws. To focus on the structural behavior of these algorithms, it is assumed that the exact value of the contact force, f_c, is available to the controllers. There are thus three basic assumptions in the following simulations for both algorithms:

(a) all mass properties are known,

(b) object and manipulator measurements, i.e. x_1, x_2, x_3, and their rates are available,

(c) the exact value of the contact force, f_c, is available.

Note that the first two assumptions are generally made in implementing most proposed algorithms in their original forms, and can be abandoned when an adaptation or parameter estimation procedure is employed. As mentioned earlier, the third assumption simplifies this comparison by eliminating the effect of a difference between contact force estimation procedures.

The system and controller parameters are

$$m_1 = 100\,kg,\ m_2 = 20.0\,kg,\ m_3 = 10.0\,kg,\ k_1 = 2.6\times10^5\ N/m,\ k_2 = 2.0\times10^4\ N/m$$

$$b_1 = 325\,kg/\sec,\ \mathrm{b}_2 = 100.0\,kg/\sec,\ m_{des} = 100.0,\ k_p = 100.0,\ k_d = 300.0$$

The initial conditions are

$$(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3)^T = (-0.2, -0.1, 0.01, 0.0, 0.0, 0.0)^T \quad (m, m/s)$$

and it is assumed that each spring is initially free of tension or compression. The desired trajectory for the object is

$$x_{3des} = 1 - e^{-t}$$

The obstacle is at $x_w = 0.7\,m$, and the contact force is computed as

$$\begin{aligned} &if \quad x_3 > x_w \quad f_c = k_w(x_w - x_3) \\ &else \quad f_c = 0.0 \end{aligned} \tag{6.40}$$

where $k_w = 1e5 \quad N/m$. To approximate actuator dynamics, the input force F_1 is filtered by a second-order Butterworth *low-pass filter*, as

$$\frac{F_{1_{filtered}}}{F_1} = \frac{\omega_0^{\,2}}{s^2 + \sqrt{2}\omega_0 s + \omega_0^{\,2}} \tag{6.41}$$

where ω_0 is chosen equal to 30 rad/sec.

Figures 6.5 and 6.6 compare the simulated performance of the MIC and OIC algorithms. As it is seen, the system never rests under the OIC law (even in 100 sec), while the MIC algorithm results in a good response. Applying the OIC law, an oscillatory error demands an oscillating input force, and consequently, the contact force oscillates, see Figure 6.6. Note that the object hits the obstacle at $t \approx 2.0$ sec. It should be mentioned that the root locus analysis shows that both OIC and MIC are stable for both "no contact" and "in contact" phases.

One may suggest that choosing larger gains or a larger desired mass parameter, can solve the problem and result in a better performance for the OIC. The simulation results of further investigations of these issues, are presented in Appendix E. It is shown that by choosing larger damping gains, k_d, the resulting oscillations for the OIC do not disappear, though the amplitudes may decrease. By choosing larger k_p's, the oscillations get worse (the amplitudes increase), while the MIC still yields a good response. Note that in this case, the root locus analysis shows that both OIC and MIC are stable in both *phases*, although the simulation indicates that the OIC becomes unstable. It is interesting to note that an on-off type nonlinear system may become unstable or experience a limit cycle, while it is switching between two linear stable modes. Longer simulation runs show that, like in previous cases, the OIC results in a limit cycle. The effect of actuator saturation limits on the performance of both algorithms was also studied.

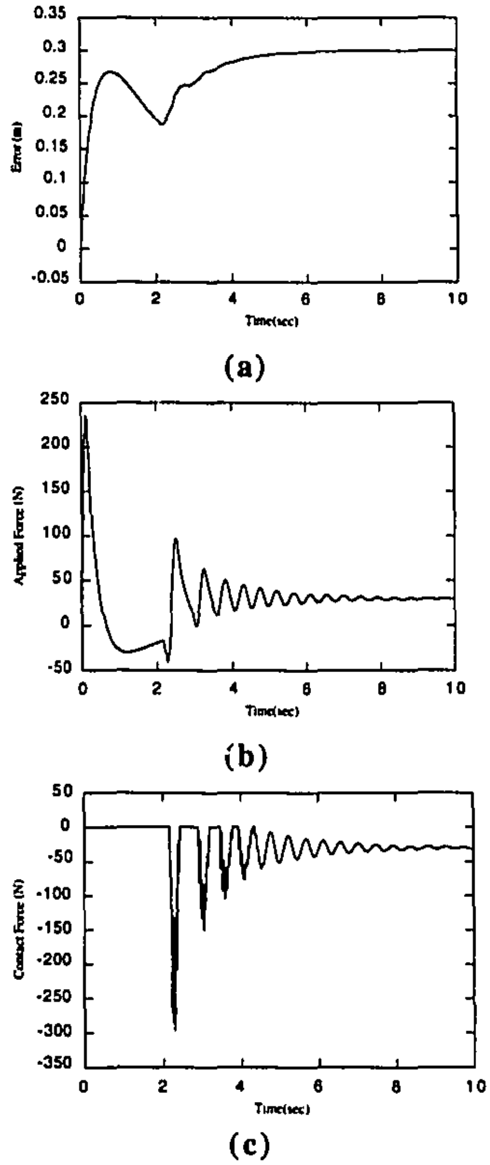


Figure 6.5: Performance of the MIC, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.

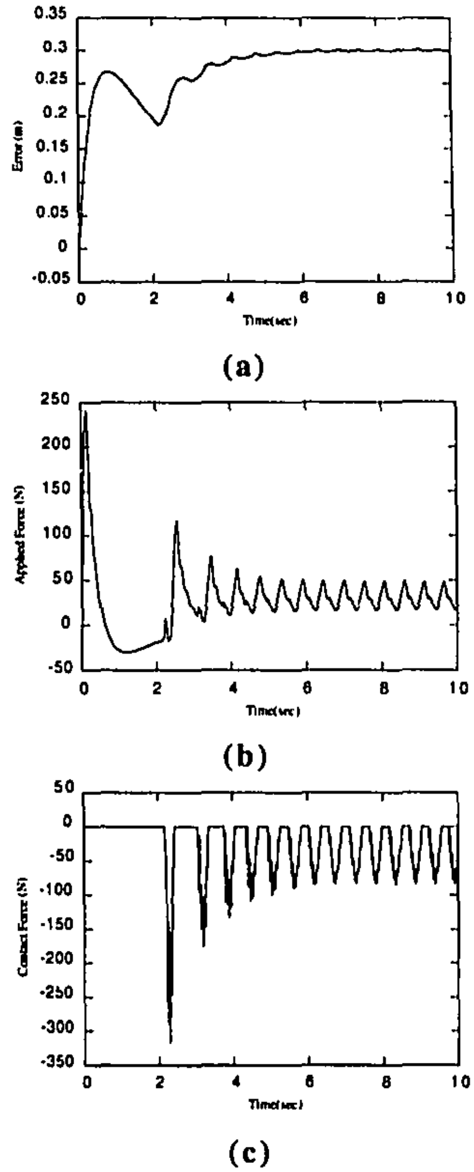


Figure 6.6: Performance of the OIC, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.

6.3.4.5 Conclusions and Discussion of the Comparative analysis

Using the linear model for a single manipulator performing an object manipulation task, it was shown with a root locus analysis that for a rigid system, both the MIC and OIC algorithms yield the same closed-loop transfer function. Also, for a rigid system in free motion (no contact with the environment) and for known values of mass properties, both algorithms yield a perfect closed-loop transfer function (i.e. G(s)=1.0). For the flexible system model, it was shown that for the OIC law, controller parameters can not affect the sum of the roots. On the contrary, the sum of the roots for the MIC algorithm is a function of controller parameters k_d and m_{des}, and this permits effective pole adjustment. The effect of choosing larger gains and larger desired mass on the stability of the OIC algorithm was investigated. It was shown that choosing larger values for the desired mass and selecting larger k_d's, can improve the stability characteristics of the OIC algorithm (see Figure D.5, Appendix D). In general, it was shown that concerning the system stability, the MIC is always preferable compared to the OIC.

Next, the performance of both algorithms was simulated and compared. To include the frequency demand of each algorithm, the input force was filtered by a second-order Butterworth low-pass filter. Also, the possibility of reaching actuator saturation limits (to exert the input force), and its effect on the performance of these algorithms was investigated (see Figures E.3-4, Appendix E). It was shown that in almost all cases, the system never rests under the OIC law, while the MIC always yields a smooth stop of the object at the obstacle. This is due to the fact that the OIC is focused on enforcing impedance law on the object motion, while the MIC is enforcing the same impedance law on both object and manipulator motions. Applying the OIC law, an oscillatory error demands an oscillating input force, and consequently, the contact force oscillates. Comparing the presented simulation results for various cases, it is concluded that the new MIC algorithm yields improved performance over the OIC.

An example application of the MIC algorithm to perform a cooperative manipulation task with two fixed-base manipulators is next presented.

6.3.5 Example: Cooperative Object Manipulation

6.3.5.1 Task Definition and Dynamics Modelling

Figure 6.7 shows a simple system of two robotic arms in planar motion, performing a *cooperative manipulation task*, i.e. moving an object with two manipulators according to predefined trajectories which may pass through an obstacle. The system includes two planar two-link manipulators each with SCARA configuration, one of which is equipped with a Remote Centre Compliance (RCC). The task is to move an object based on a given trajectory which passes through an obstacle, and the motion has to stop smoothly at the obstacle. The object has been grabbed at initial time, with a pivoted grasp condition, i.e. its orientation can change with respect to the end-effectors and no torque can be exerted on the object by any of the two end-effectors. Therefore, using the redundancy of the system, both the translational and rotational motions of the object are controlled by the end-effector *forces*.

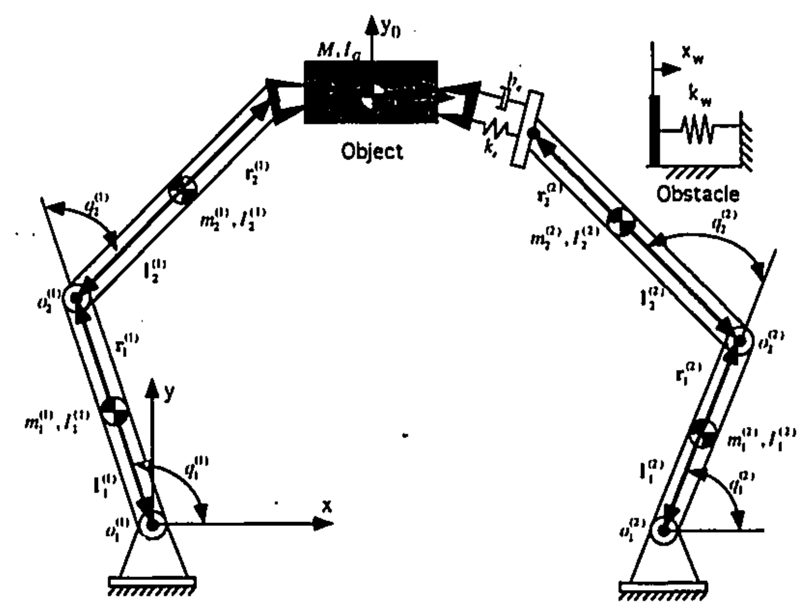


Figure 6.7: Two robotic arms, performing a cooperative manipulation task in planar motion.

Based on Eqs. (6.9) and (6.11), the system dynamics model can be represented as

$$\begin{aligned}
&\mathbf{H}^{(1)}\ddot{\mathbf{q}}^{(1)} + \mathbf{C}^{(1)} = \mathbf{Q}^{(1)} = \mathbf{J}_C^{(1)^T}\left(\tilde{\mathbf{Q}}_{app}^{(1)} - \mathbf{f}_e^{(1)}\right) \\
&\mathbf{H}^{(2)}\ddot{\mathbf{q}}^{(2)} + \mathbf{C}^{(2)} = \mathbf{Q}^{(2)} = \mathbf{J}_C^{(2)^T}\left(\tilde{\mathbf{Q}}_{app}^{(2)} - \mathbf{f}_e^{(2)}\right) \\
&m_{obj}\,\ddot{\mathbf{x}}_G = \mathbf{f}_c + \mathbf{f}_o + \mathbf{f}_e^{(1)} + \mathbf{f}_e^{(2)} \\
&I_G\,\dot{\boldsymbol{\omega}}_{obj} = \mathbf{n}_c + \mathbf{n}_o + \mathbf{r}_e^{(1)} \times \mathbf{f}_e^{(1)} + \mathbf{r}_e^{(2)} \times \mathbf{f}_e^{(2)}
\end{aligned} \tag{6.42}$$

where all terms have been already defined, see section 6.3.1. Note that the first two equations of (6.42) describe manipulator motions, and can be derived using Lagrangian approach, while the last two describe the object equations of motion. In planar motion $\boldsymbol{\omega}_{obj} = \dot{\theta}\,\hat{\mathbf{k}}$, where θ describes the object orientation with respect to xy-axis. So, the last equation can be written along z-axis, $\hat{\mathbf{k}}$, as

$$I_G\ddot{\theta} = n_c + n_o + \left(\mathbf{r}_e^{(1)} \times \mathbf{f}_e^{(1)}\right)\cdot\hat{\mathbf{k}} + \left(\mathbf{r}_e^{(2)} \times \mathbf{f}_e^{(2)}\right)\cdot\hat{\mathbf{k}} \tag{6.43}$$

where $\mathbf{n}_c = n_c\hat{\mathbf{k}}$, and $\mathbf{n}_o = n_o\hat{\mathbf{k}}$.

The kinematic constraint can be written as

$$\mathbf{x}_G = \mathbf{x}_e^{(1)} - \mathbf{r}_e^{(1)} \tag{6.44}$$

where $\mathbf{x}_e^{(1)}$ describes the first end-effector position.

To simulate the system motion, end-effector forces have to be either eliminated (e.g. using the Orthogonal Complement Method) or computed in terms of system variables. To compute these forces, first $\ddot{\mathbf{x}}_G$ can be calculated in terms of $\ddot{\mathbf{x}}_e^{(1)}$ (or $\ddot{\mathbf{q}}^{(1)}$), based on the kinematic constraint. Then, substituting the result into the object equations of motion yields

$$\mathbf{f}_e^{(1)} = \mathbf{B}_1^{-1}\left(m_{obj}\left(\dot{\mathbf{J}}_C^{(1)}\dot{\mathbf{q}}^{(1)} + \mathbf{J}_C^{(1)}\ddot{\mathbf{q}}^{(1)} + \frac{r_e^{(1)}}{I_G}(n_c + n_o)\,\mathbf{s}_1 - r_e^{(1)}\dot{\theta}^2\,\mathbf{s}_2 \right) + \mathbf{B}_2\,\mathbf{f}_e^{(2)} - \mathbf{f}_c - \mathbf{f}_o \right) \tag{6.45a}$$

where

$$\mathbf{B}_1 = \begin{bmatrix} 1 + m_{obj}(r_e^{(1)})^2\sin^2(\theta)/I_G & -m_{obj}(r_e^{(1)})^2\sin(\theta)\cos(\theta)/I_G \\ -m_{obj}(r_e^{(1)})^2\sin(\theta)\cos(\theta)/I_G & 1 + m_{obj}(r_e^{(1)})^2\cos^2(\theta)/I_G \end{bmatrix} \tag{6.45b}$$

$$\mathbf{B}_2 = \begin{bmatrix} -1 + m_{obj} r_e^{(1)} r_e^{(2)} \sin^2(\theta)/I_G & -m_{obj} r_e^{(1)} r_e^{(2)} \sin(\theta)\cos(\theta)/I_G \\ -m_{obj} r_e^{(1)} r_e^{(2)} \sin(\theta)\cos(\theta)/I_G & -1 + m_{obj} r_e^{(1)} r_e^{(2)} \cos^2(\theta)/I_G \end{bmatrix} \qquad (6.45c)$$

$$\mathbf{r}_e^{(1)} = -r_e^{(1)} \begin{Bmatrix} \cos(\theta) \\ \sin(\theta) \end{Bmatrix} \qquad \mathbf{r}_e^{(2)} = r_e^{(2)} \begin{Bmatrix} \cos(\theta) \\ \sin(\theta) \end{Bmatrix} \qquad \mathbf{s}_1 = \begin{Bmatrix} -\sin(\theta) \\ \cos(\theta) \end{Bmatrix} \qquad \mathbf{s}_2 = \begin{Bmatrix} \cos(\theta) \\ \sin(\theta) \end{Bmatrix} \qquad (6.45d)$$

and

$$\mathbf{f}_e^{(2)} = \mathbf{k}_e\left(\mathbf{x}_e^{(2)} - (\mathbf{x} + \mathbf{r}_e^{(2)}) - \mathbf{l}_{RCC}\right) + \mathbf{b}_e\left(\dot{\mathbf{x}}_e^{(2)} - (\dot{\mathbf{x}} + \dot{\mathbf{r}}_e^{(2)})\right) \qquad (6.46)$$

where $\mathbf{l}_{RCC}$ describes the RCC's free-length in different directions. Note that $\det(\mathbf{B}_1) = 1 + m_{obj}(r_e^{(1)})^2/I_G \neq 0$, therefore this matrix is always invertible and end-effector forces can be calculated as above.

The applied actuator forces, $\tilde{\mathbf{Q}}_{app}^{(1)}$ and $\tilde{\mathbf{Q}}_{app}^{(2)}$, are computed based on the MIC law as described in Eqs. (6.10, 19, 20, 23)

$$\begin{aligned} \tilde{\mathbf{Q}}_{app}^{(i)} = {} & \tilde{\mathbf{H}}^{(i)} m_{des}^{-1}\left[m_{des}\ddot{\tilde{\mathbf{x}}}_{des}^{(i)} + \mathbf{k}_d\dot{\tilde{\mathbf{e}}}^{(i)} + \mathbf{k}_p\tilde{\mathbf{e}}^{(i)} + \hat{\mathbf{F}}_c\right] + \tilde{\mathbf{C}}^{(i)} + \\ & \frac{1}{2} m_{obj} m_{des}^{-1}\left(m_{des}\ddot{\mathbf{x}}_{des} + \mathbf{k}_d\dot{\mathbf{e}} + \mathbf{k}_p\mathbf{e} + \hat{\mathbf{F}}_c\right) + \mathbf{F}_\omega - \left(\hat{\mathbf{F}}_c + \mathbf{F}_o\right) \pm \mathbf{F}_{add} \end{aligned} \qquad i = 1, 2 \quad (6.47a)$$

where $\mathbf{F}_{add}$ (with opposite sign for the two manipulators) is an additional force to create a couple (torque) by two end-effectors for controlling the object orientation in pivoted grasp condition, and

$$\begin{aligned} \mathbf{x}_{des}^{(1)} &= \mathbf{x}_{G_{des}} - \mathbf{r}_e^{(1)} \\ \mathbf{x}_{des}^{(2)} &= \mathbf{x}_{G_{des}} + \mathbf{r}_e^{(2)} - \mathbf{l}_{RCC} \end{aligned} \qquad (6.47b)$$

Next, specifying different parameters in the above equations, the system is simulated.

6.3.5.2 Simulation Results and Discussions

For the system depicted in Fig 6.7, the geometric parameters, mass properties, and the maximum available actuator torques are displayed in Table 6.1. The origin of the inertial frame is considered to be located at the fixed joint of the first manipulator. The second manipulator fixed joint is at $(1.2, 0.0)^T$. The object and controller parameters are

Table 6.1: Parameters for the system depicted in Figure 6.7.

Mani-pulator	i-th body	${}^{i}\mathbf{r}_i^{(m)}$ (m)	${}^{i}\mathbf{l}_i^{(m)}$ (m)	$m_i^{(m)}$ (kg)	$I_i^{(m)}$ (kgm^2)	$\tau_i^{(m)}$ (N-m)
1	1	0,0.50	0,-0.50	10.0	1.50	100.0
1	2	0,0.50	0,-0.50	6.0	0.80	100.0
2	1	0,0.50	0,-0.50	10.0	1.50	100.0
2	2	0,0.50	0,-0.50	8.0	0.80	100.0

$$m_{obj} = 3.0\,kg,\ I_G = 0.5\,kg\,m^2,\ {}^{0}\mathbf{r}_e^{(1)} = (-0.3,\,0.0)m,\ {}^{0}\mathbf{r}_e^{(2)} = (0.3,\,0.0)m$$

$$\mathbf{M}_{des} = diag(10,10),\ \mathbf{k}_p = diag(100,100),\ \mathbf{k}_d = diag(300,300)$$

The initial conditions are

$$(q_1^{(1)}, q_2^{(1)}, \dot{q}_1^{(1)}, \dot{q}_2^{(1)}, q_1^{(2)}, q_2^{(2)}, \dot{q}_1^{(2)}, \dot{q}_2^{(2)}, \theta, \dot{\theta})^T =$$

$$(2.7, -2.7, 0, 0, 1.0, 2.5, 0, 0, 0, 0)^T \quad (rad, rad/s)$$

and it is assumed that the RCC is initially free of tension or compression. The stiffness and damping properties for the RCC unit are chosen as, (see De Fazio, et al. (1984))

$$\mathbf{k}_e = \begin{bmatrix} 2.0\times10^4 & 0 \\ 0 & 2.0\times10^4 \end{bmatrix} kg/\sec^2, \quad \mathbf{b}_e = \begin{bmatrix} 5.0\times10^2 & 0 \\ 0 & 5.0\times10^2 \end{bmatrix} kg/\sec$$

The desired trajectory for the object center of mass, expressed in the inertial frame, is

$$x_{G_{des}} = 1 - e^{-t}\ m,\ y_{G_{des}} = 0.5\ m,\ \theta_{des} = \theta_0$$

where θ_0 describes the object initial orientation. The obstacle is at $x_w = 1.2m$, so it is expected that the object will come in contact at its right side, i.e. at $\mathbf{x}_G + \mathbf{r}_e^{(2)}$. It is assumed that no torque is developed at the contact surface (i.e. a *point contact* occurs), therefore $\mathbf{n}_c$ is equal to the moment of $\mathbf{f}_c$. Also, there is no other external force applied on the object, i.e. $\mathbf{f}_o = \mathbf{0}, \mathbf{n}_o = \mathbf{0}$. Based on these, and considering Eq. (6.43), $\mathbf{F}_{add}$ in Eq. (6.47a) is taken equal to $1/2\,\hat{\mathbf{F}}_c$ to compensate for the moment due to contact. The contact force is estimated based on Eqs. (6.29a, 30), where the real stiffness of the obstacle is

$k_w = 1e5\ N/m$. The time step which is used in the estimation procedure, Δt in Eq. (6.29a), is equal to 10 msec. Given the above information, the system is now simulated, and the obtained results are presented in Figure 6.8.

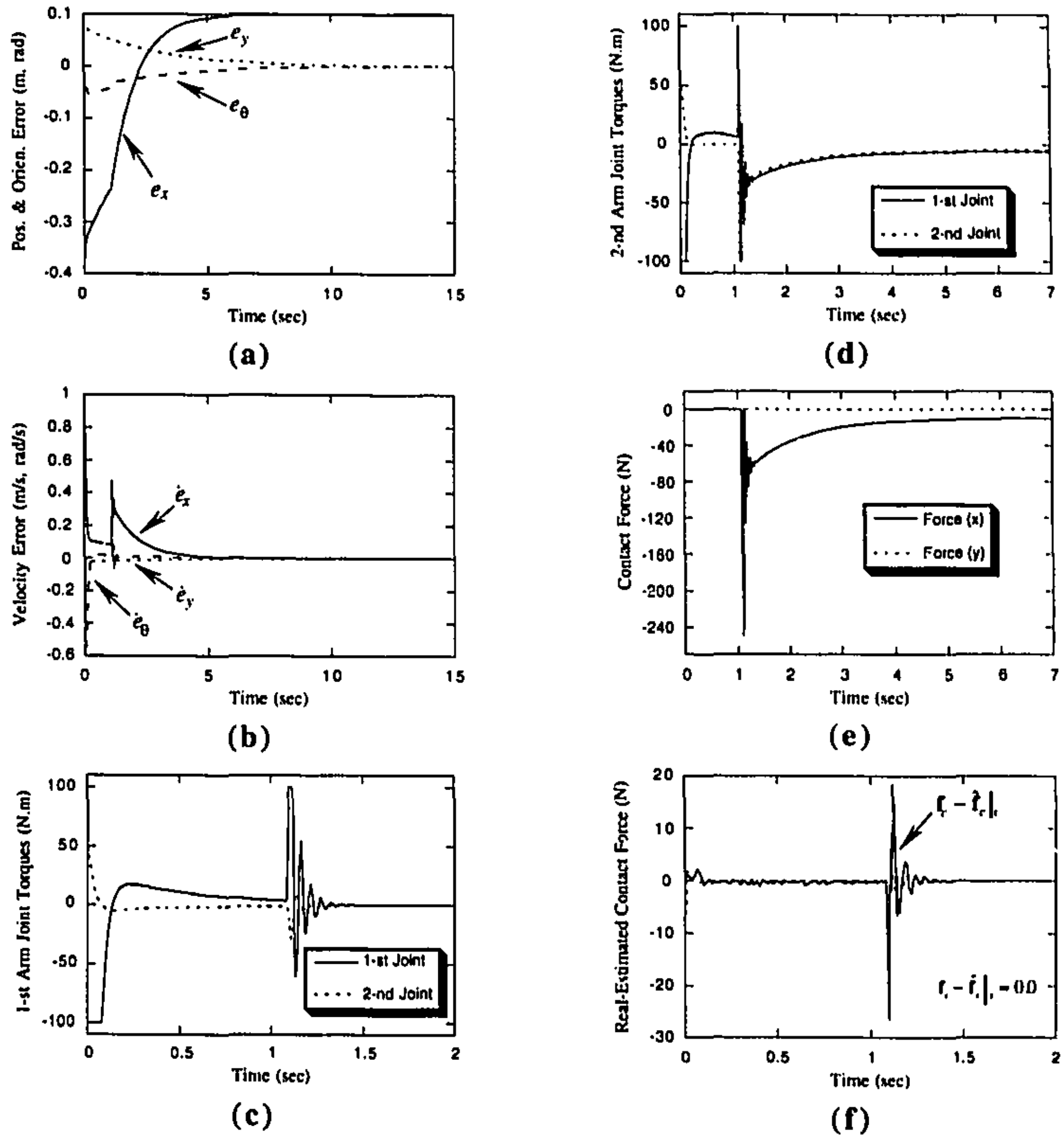


Figure 6.8: Application of the MIC in cooperative manipulation, (a) Error in object CM position and object orientation, (b) Velocity errors, (c) First manipulator joint torques, (d) Second manipulator joint torques, (e) Real value of the contact force, (f) Difference between the real value of contact force and estimated one.

As it is seen in Figures 6.8a,b the y-component of the error in the object position, starting from some initial value, converges to zero smoothly. This is due to the fact that contact occurs along the x-direction, and so the contact force does not affect the object's motion in the y-direction. The x-component of error, starting from some initial value, decreases at some rate until contact occurs at $t \approx 1.0$ sec. This rate changes after contact, because the tracking error dynamics depend on the dynamics of the environment, according to the impedance law. Then, this error smoothly converges to the distance between the final desired x-position and the obstacle x-position.

The object orientation error, starting from zero, grows to some amount and then smoothly converges to zero, Figure 6.8a. The initial growth is due to the fact that the first end-effector (i.e. without RCC) responds faster than the second one which is equipped with RCC. Therefore, according to Eq. (6.43), the difference between the two end-effector forces produces some moments which results in an undesirable rotation of the object. However, after a short transient period the difference vanishes and so does the object orientation error.

Actuator saturation limits are reached at start-up, because of large initial errors and error-rates, and at the time of hitting the obstacle, due to the contact force, Figures 6.8c,d. Joint torques for the first manipulator converge to a steady state soon after contact (about half of a second), while this takes longer for those of the second manipulator. This is due to the same reason discussed above, namely the existence of the RCC.

The contact with the obstacle occurs along the x-direction when the second end of the object passes beyond x_w. Therefore, f_{c_y} remains equal to zero before and after contact, while f_{c_x} appears whenever the object is in contact with the obstacle, Figure 6.8e. As the impact energy is dissipated, f_{c_x} converges to a constant value. According to the imposed impedance law, Eq. (6.14), for diagonal gain matrices this constant force has to be equal to $-k_p e_x = -100(0.1) = -10\,N$, which is verified from simulation results. Figure 6.8f shows the difference between real value of the contact force, and the estimated one used by the controller. As can be seen the difference is almost zero, except during a short period after impact (less than half a second). Even then, the difference is quite small (about 10% of the real value). After this period, the acceleration profiles become smoother and the difference between the real and estimated values of the contact force becomes zero. Note that before the contact, the slight difference between the two is due to the approximation of object acceleration, based on calculation of Eq. (6.29a).

Thus, simulation results show that performance of the MIC algorithm applied to a cooperative manipulation task is excellent, even in the presence of flexibility, and subject to the effects of impact with an obstacle. As described previously, different impedance parameters can be chosen for the various elements of the dynamic system when applying the MIC algorithm. However, simulation analyses (not shown here) support the physical intuition that the best results are obtained when all corresponding impedance parameters are chosen to be identical. Enforcing the same desired impedance on different parts of the system results in a harmonic accordant motion throughout the system, to achieve a good performance. Next, application of the MIC law to space robotic systems is discussed.

6.4 Application of the MIC to Space Robotics

6.4.1 Basic Formulation

The Multiple Impedance Control, as applied to a cooperative manipulation task by fixed-base manipulators, was presented in the previous section. Since for a SFFR the cooperating robotic arms are connected through a free-flying base, the implementation of the MIC algorithm has to be adapted. Here, the MIC law is formulated so as the free-flyer spacecraft exhibits the same enforced impedance as the manipulators, and the manipulated object. In fact, this is the main reason which makes the word *multiple* appear in the name of algorithm; MIC. This strategy allows compensation for an acquired object's inertia effects in the impedance law, and coordinated control of the SFFR for performing a manipulation task. It is shown that error dynamics for the spacecraft, each end-effector, and the manipulated object are following the same equation; all parts of the system are coordinately controlled based on a designated impedance law.

The vector of generalized coordinates ($\mathbf{q}$) for a multiple arm free-flyer system was defined in Eq. (3.11). The system dynamics is expressed by Eq. (3.14), repeated here as

$$\mathbf{H}(\boldsymbol{\delta}_0, \boldsymbol{\theta})\ddot{\mathbf{q}} + \mathbf{C}(\boldsymbol{\delta}_0, \dot{\boldsymbol{\delta}}_0, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \mathbf{Q}(\boldsymbol{\delta}_0, \boldsymbol{\theta}) \tag{3.14}$$

The vector of controlling variables are defined similar to Eq. (4.7), as

$$\tilde{\mathbf{x}} = [\mathbf{R}_{C_0}^T, \boldsymbol{\delta}_0^T, \mathbf{x}_E^{(1)T}, \boldsymbol{\delta}_E^{(1)T}, \cdots, \mathbf{x}_E^{(n)T}, \boldsymbol{\delta}_E^{(n)T}]^T \tag{6.48}$$

and it is assumed that all manipulators have six DOF[12], and that they all participate in manipulating the object. The vector of output speeds $\dot{\tilde{\mathbf{x}}}$ are obtained from the generalized speeds, using a square Jacobian $\mathbf{J}_C$

$$\dot{\tilde{\mathbf{x}}} = \mathbf{J}_C \, \dot{\mathbf{q}} \tag{6.49}$$

The equations of motion can now be written in the task space, i.e. in terms of the output controlled coordinates $\tilde{\mathbf{x}}$, as

$$\tilde{\mathbf{H}}(\mathbf{q})\ddot{\tilde{\mathbf{x}}} + \tilde{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \tilde{\mathbf{Q}} \tag{6.50a}$$

where

$$\tilde{\mathbf{H}} = \mathbf{J}_C^{-T} \, \mathbf{H} \, \mathbf{J}_C^{-1} \qquad \tilde{\mathbf{C}} = \mathbf{J}_C^{-T} \mathbf{C} - \tilde{\mathbf{H}} \dot{\mathbf{J}}_C \, \dot{\mathbf{q}} \qquad \tilde{\mathbf{Q}} = \mathbf{J}_C^{-T} \, \mathbf{Q} \tag{6.50b}$$

The vector of generalized forces in the task space, $\tilde{\mathbf{Q}}$, can be written similar to Eq. (6.10) for the i-th manipulator, as

$$\tilde{\mathbf{Q}} = \tilde{\mathbf{Q}}_{app} + \tilde{\mathbf{Q}}_{react} = \tilde{\mathbf{Q}}_m + \tilde{\mathbf{Q}}_f + \tilde{\mathbf{Q}}_{react} \tag{6.51}$$

where the different terms have been already defined, and will be detailed after describing the object dynamics.

12- Note that due to high safety requirements in space, a solid grasp of the object is preferred, i.e. its orientation can not change with respect to the end-effectors. So, each manipulator has to have 6 DOF.

The equations of motion for the object remain the same as those obtained in the previous section, except for the case in which the object includes an internal angular momentum source, Figure 6.9. Since this case may be of some interest for space applications, the effect of such a momentum source in the object dynamics model is discussed here.

The linear momentum of the source, $\mathbf{p}_s$, can be written as

$$\mathbf{p}_s = m_s\mathbf{v}_s = m_s(\dot{\mathbf{x}}_G + \boldsymbol{\omega}_{obj} \times \mathbf{r}_s) \tag{6.52}$$

where m_s is the mass of the angular momentum source which is not included in the object mass m_{obj}, $\mathbf{r}_s$ is position vector of the source center of mass with respect to the object CM, and $\mathbf{v}_s$ is the inertial linear velocity of the source center of mass. The required force for moving the internal angular momentum source along with the object motion, $\mathbf{F}_G$, can be written as

$$\mathbf{F}_G = \dot{\mathbf{p}}_s = \frac{d}{dt} m_s\mathbf{v}_s \tag{6.53}$$

Therefore, differentiation of Eq. (6.52) and substitution of the result into Eq. (6.53), yields

$$\mathbf{F}_G = m_s\left(\ddot{\mathbf{x}}_G + \dot{\boldsymbol{\omega}}_{obj} \times \mathbf{r}_s + \boldsymbol{\omega}_{obj} \times (\boldsymbol{\omega}_{obj} \times \mathbf{r}_s)\right) \tag{6.54}$$

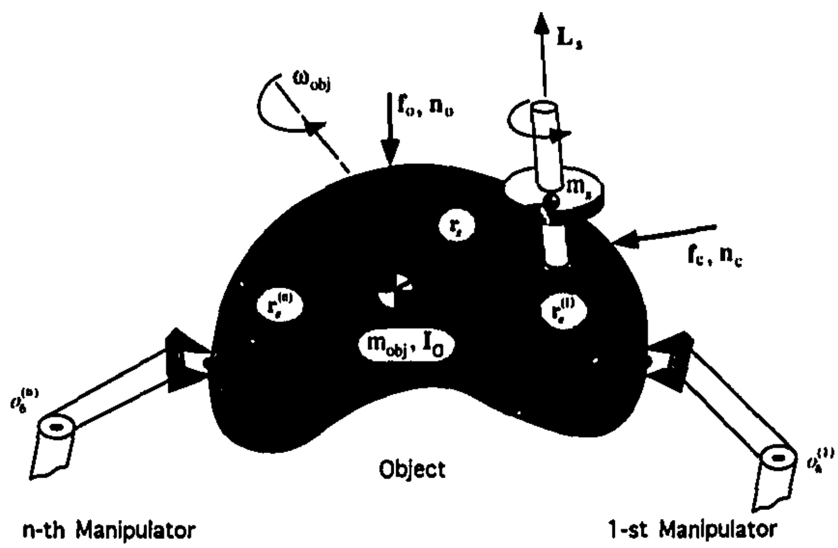


Figure 6.9: An object with an internal angular momentum source, manipulated by a multiple arm SFFR.

which has to be included in the force equilibrium equation for linear motion, Eq. (6.11), as

$$m_{obj}\ddot{\mathbf{x}}_G + m_s\left(\ddot{\mathbf{x}}_G + \dot{\boldsymbol{\omega}}_{obj}\times\mathbf{r}_s + \boldsymbol{\omega}_{obj}\times(\boldsymbol{\omega}_{obj}\times\mathbf{r}_s)\right) = \mathbf{f}_c + \mathbf{f}_o + \sum_{i=1}^{m}\mathbf{f}_e^{(i)} \qquad (6.55)$$

For the object angular motion, based on the *translation theorem* for angular momentum (Meirovitch (1970)), it can be written

$$\mathbf{L}_G = \mathbf{L}_s + \mathbf{r}_s\times\mathbf{p}_s \qquad (6.56)$$

where $\mathbf{L}_s$ is the angular momentum of the internal source about its center of mass, and $\mathbf{L}_G$ is the angular momentum of the internal source about the object center of mass. Therefore, the required moment, $\mathbf{M}_G$, for moving the internal angular momentum source along with the object motion can be written about the object center of mass as

$$\mathbf{M}_G = \dot{\mathbf{L}}_G + \dot{\mathbf{x}}_G\times\mathbf{p}_s \qquad (6.57)$$

which, based on Eqs. (6.52, 56) and assuming that $\mathbf{L}_s$ has a constant magnitude, results in

$$\mathbf{M}_G = \boldsymbol{\omega}_{obj}\times\mathbf{L}_s + \frac{d}{dt}(\mathbf{r}_s\times\mathbf{p}_s) + \dot{\mathbf{x}}_G\times m_s\left(\dot{\mathbf{x}}_G + \boldsymbol{\omega}_{obj}\times\mathbf{r}_s\right) \qquad (6.58)$$

Calculating the different terms of Eq. (6.58), and substituting the results back into the equation, yields

$$\mathbf{M}_G = \boldsymbol{\omega}_{obj}\times\mathbf{L}_s + m_s\mathbf{r}_s\times\left(\ddot{\mathbf{x}}_G + \dot{\boldsymbol{\omega}}_{obj}\times\mathbf{r}_s + \boldsymbol{\omega}_{obj}\times(\boldsymbol{\omega}_{obj}\times\mathbf{r}_s)\right) \qquad (6.59)$$

which has to be included in Eq. (6.11) for angular motion, as

$$\begin{aligned}\mathbf{I}_G\dot{\boldsymbol{\omega}}_{obj} + \boldsymbol{\omega}_{obj}\times\mathbf{I}_G\,\boldsymbol{\omega}_{obj} + \boldsymbol{\omega}_{obj}\times\mathbf{L}_s + m_s\mathbf{r}_s\times\left(\ddot{\mathbf{x}}_G + \dot{\boldsymbol{\omega}}_{obj}\times\mathbf{r}_s + \boldsymbol{\omega}_{obj}\times(\boldsymbol{\omega}_{obj}\times\mathbf{r}_s)\right)\\ = \mathbf{n}_c + \mathbf{n}_o + \sum_{i=1}^{m}\mathbf{r}_e^{(i)}\times\mathbf{f}_e^{(i)} + \sum_{i=1}^{m}\mathbf{n}_e^{(i)}\end{aligned} \qquad (6.60)$$

Similar to the general case, and following the same procedure, the object equations of motion (Eqs. (6.55, 60)) can be assembled and written in the matrix form of Eq. (6.13a), repeated here for convenience

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}_{\omega} = \mathbf{F}_c + \mathbf{F}_o + \mathbf{G}\mathbf{F}_e \qquad (6.13a)$$

where we now have

$$\mathbf{M}=\begin{bmatrix}(m_{obj}+m_s)\mathbf{1}_{3\times3} & -m_s[\mathbf{r}_s]^\times \mathbf{S}_{obj}\\ m_s\,\mathbf{S}_{obj}^T[\mathbf{r}_s]^\times & \mathbf{S}_{obj}^T(\mathbf{I}_G+\mathbf{I}_s)\mathbf{S}_{obj}\end{bmatrix} \qquad \mathbf{I}_s=m_s\begin{bmatrix} r_{s_y}^2+r_{s_z}^2 & -r_{s_x}r_{s_y} & -r_{s_x}r_{s_z}\\ -r_{s_x}r_{s_y} & r_{s_x}^2+r_{s_z}^2 & -r_{s_y}r_{s_z}\\ -r_{s_x}r_{s_z} & -r_{s_y}r_{s_z} & r_{s_x}^2+r_{s_y}^2\end{bmatrix}$$

$$\mathbf{F}_\omega=\begin{Bmatrix} m_s[\boldsymbol{\omega}_{obj}]^\times[\boldsymbol{\omega}_{obj}]^\times\mathbf{r}_s-m_s[\mathbf{r}_s]^\times\dot{\mathbf{S}}_{obj}\dot{\boldsymbol{\delta}}_{obj}\\ \mathbf{S}_{obj}^T\left([\boldsymbol{\omega}_{obj}]^\times\mathbf{I}_G\ \boldsymbol{\omega}_{obj}+[\boldsymbol{\omega}_{obj}]^\times\mathbf{L}_s+(\mathbf{I}_G+\mathbf{I}_s)\dot{\mathbf{S}}_{obj}\dot{\boldsymbol{\delta}}_{obj}+m_s[\mathbf{r}_s]^\times[\boldsymbol{\omega}_{obj}]^\times[\boldsymbol{\omega}_{obj}]^\times\mathbf{r}_s\right)\end{Bmatrix}$$

$$\mathbf{F}_c=\begin{Bmatrix}\mathbf{f}_c\\ \mathbf{S}_{obj}^T\,\mathbf{n}_c\end{Bmatrix} \qquad \mathbf{F}_o=\begin{Bmatrix}\mathbf{f}_o\\ \mathbf{S}_{obj}^T\,\mathbf{n}_o\end{Bmatrix} \qquad \mathbf{F}_e^{(i)}=\begin{Bmatrix}\mathbf{f}_e^{(i)}\\ \mathbf{n}_e^{(i)}\end{Bmatrix}_{6\times1}$$

$$\mathbf{F}_e=\begin{Bmatrix}\mathbf{F}_e^{(1)}\\ \vdots\\ \mathbf{F}_e^{(n)}\end{Bmatrix}_{6n\times1} \qquad \mathbf{G}=\begin{bmatrix}\mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \cdots & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{S}_{obj}^T[\mathbf{r}_e^{(1)}]_{3\times3}^\times & \mathbf{S}_{obj}^T & & \mathbf{S}_{obj}^T[\mathbf{r}_e^{(n)}]_{3\times3}^\times & \mathbf{S}_{obj}^T\end{bmatrix}_{6\times6n}$$

(6.61)

Note that the mass inertia matrix $\mathbf{M}$ is no longer block diagonal. Now, a desired impedance law for the object motion can be chosen as defined in Eq. (6.14)

$$\mathbf{M}_{des}\ddot{\mathbf{e}}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\mathbf{F}_c=\mathbf{0} \tag{6.14}$$

Then, following the same procedure as described for the general formulation, the required end-effector forces can be obtained as

$$\mathbf{F}_{e_{req}}=\mathbf{G}^{\#}\left\{\mathbf{M}\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\mathbf{F}_c\right)+\mathbf{F}_\omega-\left(\mathbf{F}_c+\mathbf{F}_o\right)\right\} \tag{6.62}$$

or (substituting the estimated contact force for the actual one)

$$\mathbf{F}_{e_{req}}=\mathbf{G}^{\#}\left\{\mathbf{M}\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\hat{\mathbf{F}}_c\right)+\mathbf{F}_\omega-\left(\hat{\mathbf{F}}_c+\mathbf{F}_o\right)\right\} \tag{6.63}$$

where $\mathbf{G}^{\#}$ is the pseudoinverse of the grasp matrix $\mathbf{G}$, a full-rank matrix defined by Eq. (6.16b). Note that in space operations it is preferred to grab a targeted object with a special tool or grippers. Therefore, there is no requirement to produce internal forces and moments in the object and, compared to Eq. (6.18), $\mathbf{F}_{\text{int}}$ is chosen to be zero. Then, considering Eq. (6.51), the controlled force required to be applied on the manipulated object by the end-effectors is

$$\tilde{\mathbf{Q}}_f = \begin{Bmatrix} \mathbf{0}_{6\times1} \\ \mathbf{F}_{e_{req}} \end{Bmatrix} \tag{6.64}$$

and, similar to the general case, the reaction load on the system is

$$\tilde{\mathbf{Q}}_{react} = \begin{Bmatrix} \mathbf{0}_{6\times1} \\ -\mathbf{F}_e \end{Bmatrix} \tag{6.65a}$$

where

$$\mathbf{F}_e = \mathbf{G}^{\#}[\mathbf{M}\ddot{\mathbf{x}} + \mathbf{F}_{\omega} - (\mathbf{F}_c + \mathbf{F}_o)] \tag{6.65b}$$

Next, to complete the computation of the controlling force, as described in Eq. (6.51), a proper expression for $\tilde{\mathbf{Q}}_m$ must be obtained.

To impose the same impedance law on the spacecraft motion, manipulators, and the manipulated object, the impedance law for the space free-flyer can be written as

$$\tilde{\mathbf{M}}_{des}\ddot{\tilde{\mathbf{e}}} + \tilde{\mathbf{k}}_d\dot{\tilde{\mathbf{e}}} + \tilde{\mathbf{k}}_p\tilde{\mathbf{e}} + \mathbf{U}_{f_c}\mathbf{F}_c = \mathbf{0}_{N\times1} \tag{6.66a}$$

where $\tilde{\mathbf{e}} = \tilde{\mathbf{x}}_{des} \cdot \tilde{\mathbf{x}}$ is the tracking error in the SFFR controlled variables (note that e describes the tracking error in the object position and orientation), $\mathbf{U}_{f_c} = [\mathbf{1}_{6\times6} \ \cdots \ \mathbf{1}_{6\times6}]^T$ is an $N \times 6$ matrix, and

$$\tilde{\mathbf{k}}_p = \begin{bmatrix} \mathbf{k}_p & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_p & \cdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{k}_p \end{bmatrix}_{N\times N} \qquad \tilde{\mathbf{k}}_d = \begin{bmatrix} \mathbf{k}_d & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{k}_d & \cdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{k}_d \end{bmatrix}_{N\times N}$$

$$\tilde{\mathbf{M}}_{des} = \begin{bmatrix} \mathbf{M}_{des} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{des} & \cdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}_{des} \end{bmatrix}_{N\times N} \tag{6.66b}$$

and N = 6n+6 is the SFFR total DOF. Note that the desired trajectory for the system controlled variables, $\tilde{\mathbf{x}}_{des}$, can be defined based on the desired trajectory for the object motion, $\mathbf{x}_{des}$, and the grasp condition, as discussed in the general formulation. Then, similar to the above derivation for $\tilde{\mathbf{Q}}_f$, $\tilde{\mathbf{Q}}_m$ can be obtained as

$$\tilde{\mathbf{Q}}_m = \tilde{\mathbf{H}}(\mathbf{q})\tilde{\mathbf{M}}_{des}^{-1}\left[\tilde{\mathbf{M}}_{des}\ddot{\tilde{\mathbf{x}}}_{des}+\tilde{\mathbf{k}}_d\dot{\tilde{\mathbf{e}}}+\tilde{\mathbf{k}}_p\tilde{\mathbf{e}}+\mathbf{U}_{f_c}\mathbf{F}_c\right]+\tilde{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}}) \tag{6.67a}$$

or, substituting the estimated value for the contact force

$$\tilde{\mathbf{Q}}_m = \tilde{\mathbf{H}}(\mathbf{q})\tilde{\mathbf{M}}_{des}^{-1}\left[\tilde{\mathbf{M}}_{des}\ddot{\tilde{\mathbf{x}}}_{des}+\tilde{\mathbf{k}}_d\dot{\tilde{\mathbf{e}}}+\tilde{\mathbf{k}}_p\tilde{\mathbf{e}}+\mathbf{U}_{f_c}\hat{\mathbf{F}}_c\right]+\tilde{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}}) \tag{6.67b}$$

where $\tilde{\mathbf{M}}_{des}^{-1}$ can be computed as

$$\tilde{\mathbf{M}}_{des}^{-1} = \begin{bmatrix} \mathbf{M}_{des}^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{des}^{-1} & \cdots & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}_{des}^{-1} \end{bmatrix}_{\mathrm{N}\times\mathrm{N}} \tag{6.67c}$$

Next, it is shown that the error dynamics for the spacecraft, each end-effector, and the manipulated object are expressed by the same equation.

6.4.2 Error Analysis

Substituting Eqs. (6.67b), (6.65a), and (6.64) into Eq. (6.51), and the result into Eq. (6.50a) yields

$$\tilde{\mathbf{H}}(\mathbf{q})\left(\tilde{\mathbf{M}}_{des}^{-1}\left(\tilde{\mathbf{M}}_{des}\ddot{\tilde{\mathbf{x}}}_{des}+\tilde{\mathbf{k}}_d\dot{\tilde{\mathbf{e}}}+\tilde{\mathbf{k}}_p\tilde{\mathbf{e}}+\mathbf{U}_{f_c}\mathbf{F}_c\right)-\ddot{\tilde{\mathbf{x}}}\right)+ \left\{\begin{matrix} \mathbf{0}_{6\times1} \\ \mathbf{G}^{\#}\mathbf{M}\left(\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\mathbf{F}_c\right)-\ddot{\mathbf{x}}\right)\end{matrix}\right\}=\mathbf{0} \tag{6.68}$$

where it is assumed that the exact value of the contact force is available, also mass and geometric properties for the manipulated object, and spacecraft/manipulating system are given. Since Eq. (6.68) must hold for any $\mathbf{M}$ and any $\tilde{\mathbf{H}}$, it can be concluded that

$$\begin{aligned} &\tilde{\mathbf{H}}(\mathbf{q})\left(\tilde{\mathbf{M}}_{des}^{-1}\left(\tilde{\mathbf{M}}_{des}\ddot{\tilde{\mathbf{x}}}_{des}+\tilde{\mathbf{k}}_d\dot{\tilde{\mathbf{e}}}+\tilde{\mathbf{k}}_p\tilde{\mathbf{e}}+\mathbf{U}_{f_c}\mathbf{F}_c\right)-\ddot{\tilde{\mathbf{x}}}\right)=\mathbf{0} \\ &\mathbf{G}^{\#}\mathbf{M}\left(\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\mathbf{F}_c\right)-\ddot{\mathbf{x}}\right)=\mathbf{0} \end{aligned} \tag{6.69}$$

and, since $\mathbf{G}^{\#}$ is of full-rank, this results in

$$
\begin{aligned}
&\tilde{\mathbf{H}}(\mathbf{q})\left(\tilde{\mathbf{M}}_{des}^{-1}\left(\tilde{\mathbf{M}}_{des}\ddot{\tilde{\mathbf{x}}}_{des}+\tilde{\mathbf{k}}_d\dot{\tilde{\mathbf{e}}}+\tilde{\mathbf{k}}_p\tilde{\mathbf{e}}+\mathbf{U}_{f_c}\mathbf{F}_c\right)-\ddot{\tilde{\mathbf{x}}}\right)=\mathbf{0} \\
&\mathbf{M}\left(\mathbf{M}_{des}^{-1}\left(\mathbf{M}_{des}\ddot{\mathbf{x}}_{des}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\mathbf{F}_c\right)-\ddot{\mathbf{x}}\right)=\mathbf{0}
\end{aligned}
\tag{6.70}
$$

Finally, based on the fact that $\mathbf{M}$ and $\tilde{\mathbf{H}}$ are positive definite inertia matrices, Eq. (6.70) results in

$$
\begin{aligned}
&\tilde{\mathbf{M}}_{des}\ddot{\tilde{\mathbf{e}}}+\tilde{\mathbf{k}}_d\dot{\tilde{\mathbf{e}}}+\tilde{\mathbf{k}}_p\tilde{\mathbf{e}}+\mathbf{U}_{f_c}\mathbf{F}_c=\mathbf{0} \\
&\mathbf{M}_{des}\ddot{\mathbf{e}}+\mathbf{k}_d\dot{\mathbf{e}}+\mathbf{k}_p\mathbf{e}+\mathbf{F}_c=\mathbf{0}
\end{aligned}
\tag{6.71}
$$

Considering the definitions for $\tilde{\mathbf{M}}_{des}$, $\tilde{\mathbf{k}}_d$, $\tilde{\mathbf{k}}_p$, and $\mathbf{U}_{f_c}$ as described in Eq. (6.66b), Eq. (6.71) means that all participating manipulators, the free-flyer-base, and the manipulated object exhibit the same impedance behavior. This guarantees an accordant motion of different parts of the system for performing manipulation tasks.

6.5 Concluding Remarks

In this chapter, a new algorithm called Multiple Impedance Control (MIC), was developed. The MIC enforces a controlled impedance on cooperating manipulators *and* on the manipulated object, which results in a harmony between different parts of the system. Similar to the standard impedance control, one of the benefits of this algorithm is the ability to perform both free motions and contact tasks without switching the control modes. In addition, an object's inertia effects are compensated in the impedance law, and at the same time, the end-effector(s) tracking errors are controlled.

To reveal the merits of this new algorithm, a conceptual comparative analysis between different control strategies was first presented. Then, a general formulation for the MIC algorithm was derived, and it was shown by error analysis that under the MIC law all participating manipulators, and the manipulated object exhibit the same controlled impedance behavior. An estimation procedure for contact force determination was given which results in a good approximation, even during contact.

A linear model of an object manipulation task by a single manipulator was considered to present a thorough comparative analysis between the MIC and Object Impedance Control (OIC). A root locus analysis was used to investigate the stability of both algorithms. It was shown that for a rigid system, both algorithms yield the same closed-loop transfer function. However, in the presence of flexibility, it was shown that the MIC algorithm has superior stability properties. A simulation was used to demonstrate that the system may never rest under the OIC law, while the MIC algorithm results in a good performance. Application of the MIC law to a system of two cooperating two-link manipulators was also simulated. As simulation results revealed, even in the presence of flexibility and impact forces due to hitting an obstacle, the performance of the MIC algorithm is reasonably smooth and highly acceptable.

Finally, application of the MIC law to space robotic systems was formulated. In space, participating robotic arms are connected through a free-flying base, and the general formulation was adapted to consider the dynamic coupling between the arms and the base. For the manipulated object, inclusion of an internal source of angular momentum was admitted. By error analysis it was shown that, under the MIC law, all participating manipulators the free-flyer base, and the manipulated object exhibit the same designated impedance behavior; resulting in an accordant motion throughout the system for performing the task.

Chapter 7

Conclusions and Suggestions

7.1 Conclusions

This thesis deals with dynamics and coordinated control of multiple manipulator SFFR for the capture and manipulation of space objects. Two basic approaches for kinematics modelling of such systems were developed in Chapter 2. The barycentric vector approach was developed based on taking the system CM as a representative point for the system's translational motion, and on using a set of body-fixed barycentric vectors which reflect both geometric configuration and mass distribution of the system. This approach results in decoupling the total linear and angular motion from the rest of the equations, when no external forces/torques are applied on the system. On the other hand, the direct path approach was developed based on taking a point on the spacecraft (preferably its CM) as a representative point for the system's translation, and on using a set of body-fixed geometric vectors. Comparing the results, it was seen that the direct path approach yields considerably more compact relationships. This seems a more appropriate approach when dealing with multiple arm systems, especially when there are some external forces and torques acting on the system.

In Chapter 3, based on the developed kinematics approaches, the general Lagrangian formulation was applied to obtain the dynamics model of a space robotic system. Based on the preliminary results, the direct path approach was chosen to develop a concise explicit dynamics model of multiple manipulator SFFR in free-flying mode. Note that to develop model-based algorithms for controlling a free-floating system, the obtained dynamics model has to be reduced by mathematical techniques such as the Orthogonal Complement Method. However, if the barycentric vector approach is used, the equations can be directly reduced and employed for such a purpose. Next, a quasi-coordinate formulation of the system dynamics, and a formulation using Euler parameters for orientation representation were presented. Also, specific characteristics of space robotic systems, compared to fixed-base manipulators, were discussed. It was shown that any deviation in the estimated values of mass parameters has a drastic effect on the performance of model-based controllers in space. Finally, the symbolic programming of the dynamics equations was compared to a numerical routine, and the generation of the dynamics code was described.

The coordination between a spacecraft motion and its several end-effectors to capture a moving space object was investigated in Chapter 4. Appropriate trajectories for the spacecraft and its manipulators were planned to result in a smooth capture of moving objects in space. To perform the task, two model-based control algorithms, based on an Euler angle (MB1) and on an Euler parameter description of the orientation (MB2), and a transpose Jacobian control algorithm (TJ) were developed. The MB1 presents the inconvenience of representational singularities due to Euler angle description of the orientation, while the MB2 overcomes these non-physical singularities. Multiple arm free-flying systems were simulated, in both planar and 3-dimensional maneuvers, to investigate various aspects of the trajectory planning strategy, and to compare the performance of the developed algorithms. It was shown that a symmetric grasp results in reduced disturbances on the spacecraft. Also, for a given maneuver duration, by choosing the maximum deceleration smaller than the maximum acceleration, a smoother operation can be obtained. It was shown that if dynamic properties are accurately known, model-based controllers provide good tracking, but are computationally expensive. However, due to the complexity of the dynamics of space robotic systems, the performance of these algorithms deteriorates in the presence of higher levels of model uncertainties. On the other hand, the simple transpose Jacobian algorithm, when used with high gains, provides an acceptable and computationally inexpensive controller. However, in practice the use of very high gains are limited due to the presence of noise and unmodelled dynamics. Therefore, further work on the TJ algorithm was motivated, aiming at improving its characteristics as a good candidate for space applications.

The Modified Transpose Jacobian (MTJ) algorithm was presented in Chapter 5. Employing stored data of the previous time step control command, this algorithm yields an improved performance in terms of tracking errors, over the standard one. This new algorithm approximates a feedback linearization solution, with no need to a priori knowledge of the plant dynamics. Therefore, unlike a model-based algorithm, it is not affected by modelling inaccuracies and uncertainties. It was shown by simulation that its performance is comparable to that of model-based algorithms, and has the advantage that it requires reduced computational effort. Unlike the standard TJ, this algorithm works well in high speed tracking tasks. In addition, controller gains can be selected in a more systematic manner rather than in a heuristic way, and the noise rejection characteristics of the algorithm are improved. The new MTJ algorithm is recommended for all applications, particularly for motion control of space robotic systems, where computational power is limited yet relatively high precession is demanded.

To manipulate a captured object by multiple manipulators, both end-effector motions and forces have to be considered. To this end, the new Multiple Impedance Control (MIC) was developed in Chapter 6. The presented algorithm enforces a controlled impedance on each participating manipulator, and on the manipulated object. This algorithm can be employed for both free motions and contact tasks without switching the control modes. After a conceptual comparative analysis between different control strategies, the general formulation of the MIC algorithm was developed. It was shown that under the MIC law, all participating manipulators and the manipulated object exhibit the same pre-set impedance behavior. Therefore, a harmonic accordant motion of different parts of the system is obtained which leads to a good system performance. Discussing the similarities and differences between the MIC and the Object Impedance Control (OIC), a simple model of performing object manipulation task by a single manipulator was considered to compare the two algorithms. It was shown that in the presence of flexibility, the system does not rest under the OIC law (either becomes unstable or enters a limit cycle), while the MIC algorithm results in a well damped response and smooth stop of the object at the obstacle. Next, a system of two cooperating two-link manipulators was simulated, where a Remote Centre Compliance was attached to the second end-effector. As shown by simulation, even with flexible elements and an impact due to hitting an obstacle, the performance of the MIC algorithm was reasonably good and reliable. Finally, the MIC law was applied to a multiple arm space robotic system, where the dynamic coupling between the arms and the base, and an internal angular momentum source for the object were taken into account. It was shown that under the MIC law, the participating manipulators, the free-flying spacecraft, and the manipulated object exhibit the same controlled impedance behavior. This strategy permits coordinated control of a multiple manipulator SFFR in performing a manipulation task, as well as compensation for an acquired object's inertia effects in the impedance law.

7.2 Suggestions for Future Research

In this research work, the dynamics and control of multiple arm space robotic systems was studied. To extend the obtained results, and develop new contributions to this fast growing field of science, some suggestions for further research are presented in the following.

Non-square Jacobians. In developing the control algorithms, presented in this study, it was assumed that the system is sufficiently actuated. In other words, the vector of actuator forces/toques was assumed to be related to the vector of generalized forces by an N×N square Jacobian matrix, where N is the system DOF. Formulating the developed algorithms for non-square Jacobians, inspired by the objectives of controlling over-actuated or under-actuated systems, would be interesting for further research.

On-off thrusters. Controlling space manipulators, in free-flying mode, requires applying spacecraft thruster forces/torques. Application of the developed control algorithms in simulated systems was based on the assumption that actuator forces/torques, including those exerted by thrusters, are continuous. However, current space technology uses compressed-gas on-off thrusters, to avoid valve clogging from freezing. Although space technology is developing fast, and this may not be a problem in near future, considering on-off thrusters will yield more realistic results.

An MIC algorithm for several free-flying robots. The new MIC law, for space applications, was developed for a multiple arm free-flyer system, assuming that each manipulator has six DOF and all participate in manipulating the object. The fact that some appendages may not participate in performing the task, can be easily included in this formulation. Also, admitting extra DOF for activated manipulators can be helpful for implementation of the algorithm in redundant systems. Development of the MIC law for a centralized control of several free-flying robots in manipulating an object, can be pursued based on the same structure as implemented in cooperation of several manipulators. This is another interesting subject for further research.

An MIC law with no requirement of manipulator dynamics knowledge. As shown in this study the performance of model-based algorithms, in space, is more affected by the accuracy in the estimation of mass parameters. Similar to model-based algorithms, the MIC law requires knowledge of the system dynamics concerning the manipulators motion (in computation of $\tilde{\mathbf{Q}}_m$). Therefore, it is an interesting subject for future research to substitute the motion-concerned part of the MIC law, with the developed MTJ algorithm which does not require any priori knowledge of the system dynamics.

Design aspects. The dynamics generation and simulation codes were used to evaluate the performance of alternative developed control algorithms. These codes can be used in the design procedure of space free-flying robotic systems in both architecture, and functionality. Further research can be done in this area, to develop some useful design guidelines.

Experimental studies. In this research work, simulation routines were very helpful to improve the new algorithms, and evaluate them, where a graphical simulation code was used to obtain an animated picture of the whole maneuver. However, experimental studies can show the merits of the developed theories in a real implementation, and may bring up some hidden points to improve the presented algorithms.

Bibliography

Agrawal, S. K. and Desmier, G., "Models of Assembly Primitives for Free-Floating Robots," *Proc. of the Int. Conf. on Intelligent Robots and Systems*, IROS'93, Yokohama, Japan, July 1993, pp. 2041-2048.

Alexander, H. and Cannon, R., "An Extended Operational-Space Control Algorithm for Satellite Manipulators," *The Journal of the Astronautical Sciences*, Vol. 38, No. 4, October-December 1990, pp. 473-486.

An, C. H., Atkeson, C. G., Griffiths, J. D., and Hollerbach, J. M., "Experimental Evaluation of Feedforward and Computed Torque Control," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, June 1989, pp. 368-373.

Angeles, J., *Rational Kinematics*, Springer-Verlag, New York, 1988.

Arimoto, S. and Miyazaki, F., "Stability and Robustness of PID Feedback Control for Robot Manipulators of Sensory Capability," *Proc. First Int. Symp. on Robotics Research*, Cambridge, Massachusetts 1983, pp. 783-799.

Arimoto, S. and Miyazaki, F., "Can Mechanical Robots Learn by Themselves?," *Proc. Second Int. Symp. on Robotics Research*, Kyoto, Japan, 1984, pp. 127-134.

Asada, H. and Slotine, J. J., "*Robot Analysis and Control*," Wiley Interscience, 1986.

Asari, Y., Sato, H., Yoshimi, T., and Tatsuno, K., "Development of a model-based remote maintenance robot system (IV) - a practical stiffness control method for redundant robot arm," *Proc. Int. Conf. on Intelligent Robots and Systems*, IROS'93, Yokohama, Japan, 1993.

Bejczy, A., "Robot Arm Dynamics and Control," *Jet Propulsion Laboratory Technical Memo 33-669*, February 1973.

Bronez, M. A., Clarke, M. M., and Quinn, A., "Requirements Development for a Free-Flying Robot - the ROBIN," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, April 1986.

Carusone, J., Buchan, K. S., and D'Eleuterio, G. M. T., "Experiments in End-Effector Tracking Control for Structurally Flexible Space Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, October 1993, pp. 553-560.

Chiaverini, S., Sciavicco, L., and Siciliano, B., "Control of robotic systems through singularities," *Proc. of the Int. Workshop on Nonlinear and Adaptive Control: Issues in Robotics*, Grenoble, France, 1990. Also in *Lecture Notes in Control & Information Sciences*, v. 162, 1991.

Craig, J., *Introduction to Robotics, Mechanics and Control*, Addison Wesley, Reading, MA, 1989.

Cyril, X., Angeles, J., and Misra, A., "Dynamics of Flexible Multibody Mechanical Systems," *Transactions of the CSME*, Vol. 15, No. 3, 1991, pp. 235-256.

De Fazio, T. L., Seltzer, D. S., and Whitney, D. E., "The Instrumented Remote Centre Compliance," *Journal of The Industrial Robot*, Vol. 11, No. 4, December 1984, pp. 238-242.

Dickson, C. and Cannon, R. H., "Experimental Results of Two Free-Flying Robots Capturing and Manipulating a Free-Flying Object," *Proc. of the Int. Conf. on Intelligent Robots and Systems*, IROS'95, Pittsburgh, PA, August 1995, pp. 51-58.

Dubowsky, S. and Des Forges, D. T., "The Application of Model-Referenced Adaptive Control to Robotic Manipulators," *ASME Journal of Dynamic Systems, Measurement & Control*, Vol. 101, 1979.

Dubowsky, S. and Papadopoulos, E., "The Dynamics and Control of Space Robotic Systems," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, October 1993, pp. 531-543.

Dubowsky, S. and Torres, M., "Path Planning for Space Manipulators to Minimize Spacecraft Attitude Disturbances," *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 1991, pp. 2522-2528.

Fujii H., Murayama T., Nakajima K., and Anazawa S., "Capture Control for Manipulator Arm of Free-flying Space Robot," AIAA-90-3432-CP.

Goldenberg, A. A., "Implementation of Force and Impedance Control in Robot Manipulators," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Philadelphia, Pennsylvania, April 1988, pp. 1626-1632.

Hayati, S., "Hybrid Position/Force Control of Multi-Arm Cooperating Robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, April 1986, pp. 82-89.

Hayward, V. and Hayati, S., "KALI: An Environment for the Programming and Control of Cooperative Manipulators," *Proc. of American Cont. Conf.*, Piscataway, NJ, USA, 1988, pp. 473-478.

Hogan, N., "Impedance Control: An Approach to Manipulation -A Three Part Paper," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 107, March 1985, pp. 1-24.

Hootsmans, N. A. M. and Dubowsky, S., "Large motion control of mobile manipulators including vehicle suspension characteristics," *Proc. IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, USA, April 1991.

Hughes, P.C., *Spacecraft Attitude Dynamics*, John Wiley, New York, NY, 1986.

Jaar, G., Cyril, X., and Misra, A. K., "Dynamical Modelling and Control of a Spacecraft-Mounted Manipulator Capturing a Spinning Satellite," *43rd Congress of The Int. Astronautical Federation*, September 1992, IAF-92-0029.

Ju, M. S. and Mansour, J. M., "Comparison of Methods for Developing the Dynamics of Rigid-Body Systems," *The International Journal of Robotics Research*, Vol. 8, No. 6, December 1989, pp. 19-27.

Kane, T. R. and Levinson, D. A., *Dynamics: Theory and Applications*, McGraw-Hill, 1985.

Kawamura, S., Miyazaki, F., and Arimoto, S., "Is a Local Linear PD Feedback Control Law effective for Trajectory Tracking of Robot Motion?," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Philadelphia, Pennsylvania, April 1988, pp. 1335-1340.

Khatib, O., "A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation," *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 1, February 1987, pp. 43-53.

Khosla, P. K. and Kanade, T., "Real-Time Implementation and Evaluation of Computed Torque Scheme," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 2, April 1989, pp. 245-253.

Leahy, M. B. Jr. and Saridis, G. N., "Compensation of Industrial manipulator Dynamics," *The International Journal of Robotics Research*, Vol. 8, No. 4, August 1989, pp. 73-84.

Markiewicz, B., "Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computed-Controlled Manipulator," *Jet Propulsion Laboratory Technical Memo 33-601*, March 1973.

Meer, D. W. and Rock, S. M., "Coupled-System Stability of Flexible-Object Impedance Control," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995, pp. 1839-1845.

Meirovitch, L., *Methods of Analytical Dynamics*, McGraw-Hill, 1970.

Miyazaki, F., Masutani, Y., and Arimoto, S., "Sensor feedback using approximate Jacobian," *Proc. of the USA-Japan Symposium on Flexible Automation - Crossing Bridges*, Advances in Flexible Automation and Robotics, Minneapolis, MN, USA, 1988, pp. 139-145.

Mukherjee, R. and Chen, D., "Control of Free-Flying Underactuated Space Manipulators to Equilibrium Manifolds," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, October 1993, pp. 561-570.

Nagamatsu, H., Kubota, T., and Nakatani, I., "Capture Strategy for Retrieval of a Tumbling Satellite by a Space Robotic Manipulator," *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 1996, pp. 70-75.

Nakamura, Y., Nagai, K., and Yoshikawa, T., "Mechanics of Coordinative Manipulation by Multiple Robotic Mechanisms," *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 1987, pp. 991-998.

Nakamura, Y. and Mukherjee, R., "Exploiting Nonholonomic Redundancy of Free-Flying Space Robots" *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 4, August 1993, pp. 499-506.

Ogata, K., *Modern Control Engineering*, Prentice Hall, 1970.

Paielli, R. A. and Bach, R. E., "Attitude Control with Realization of Linear Error Dynamics," *J. of Guidance, Control, and Dynamics*, Vol. 16, No. 1, 1993, pp. 182-189.

Papadopoulos, E. and Dubowsky, S., "On The Nature of Control Algorithms for Free-Floating Space Manipulators," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 6, December 1991a, pp. 750-758.

Papadopoulos, E. and Dubowsky, S., "Coordinated Manipulator/Spacecraft Motion Control for Space Robotic Systems," *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 1991b, pp. 1696-1701.

Papadopoulos, E. and Moosavian, S. Ali A., "Dynamics & Control of Multi-arm Space Robots During Chase & Capture Operations," *Proc. Int. Conf. on Intelligent Robots and Systems (IROS '94)*, Munich, Germany, Sept. 12-16, 1994a.

Papadopoulos, E. and Moosavian, S. Ali A., "Trajectory Planning & Control of Multiple Arm Space Free-Flyers," *Proc. of the 2nd Workshop on Robotics in Space*, International Advanced Robotics Program, Montreal, Canada, July 6-8, 1994b.

Papadopoulos, E. and Moosavian, S. Ali A., "A Comparison of Motion Control Algorithms for Space Free-flyers," *Proc. of the 5th Int. Conf. on Adaptive Structures*, Sendai, Japan, December 5-7, 1994c.

Papadopoulos, E. and Moosavian, S. Ali A., "Dynamics & Control of Space Free-Flyers with Multiple Arms," *Journal of Advanced Robotics*, Vol. 9, No. 6, 1995, pp. 603-624.

Paul, R. P., "Modelling Trajectory Calculation, and Servoing of a Computer Controlled Arm," *Technical Report AIM-177*, Stanford University Artificial Intelligence Laboratory, 1972.

Raibert, M. H. and Craig, J. J., "Hybrid Position/Force Control of Manipulators," *ASME Journal of Dynamic Systems, Measurement & Control*, Vol. 126, June 1981, pp. 126-133.

Reuter, G. J., et al. "An Intelligent, Free-Flying Robot, in Space Station Automation IV," Chiou, W. C. ed., *SPIE Proceedings Series*, Vol. 1006, November 1988, pp. 20-27.

Rivin, E. I., *Mechanical Design of Robots*, McGraw-Hill, 1988.

Saha, S. K. and Angeles, J., "Dynamics of Nonholonomic Mechanical Systems Using a Natural Orthogonal Complement," *ASME Journal of Applied Mechanics*, Vol. 58, March 1991, pp. 238-244.

Schneider, S. A. and Cannon, R. H., "Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 3, June 1992, pp. 383-394.

Seraji, H. and Colbaugh, R., "Force Tracking in Impedance Control," *Proc. of IEEE Int. Conf. on Robotics and Automation*, Atlanta, Georgia, May 1993, pp. 499-506.

Slotine, J. J. E. and Li, W., "Adaptive Robot Control: A New Perspective," *Proc. of the 26th IEEE Conf. on Decision and Control*, Los Angeles, CA, USA, 1987.

Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Prentice Hall, 1991.

Strang, G., *Linear Algebra and its Applications*, Harcourt Brace Javanovitch Publishers, 1988.

Takahashi, Y., *Control and Dynamic Systems*, Addison-Wesley Publication Co., 1970.

Tarn, T. J., Bejczy, A. K., and Yun, X., "Design of Dynamic Control of Two Cooperating Robot Arms: Closed Chain Formulation," *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 1987, pp. 7-13.

Ullman, M. and Cannon, R. H., "Experiments in Global Navigation and Control of a Free-Flying Space Robot," *Proc. of ASME WAM* (San Francisco), December 1989, ASME DSC-vol. 15, pp. 37-43.

Umetani, Y. and Yoshida, K., "Continuous Path Control of Space Manipulators Mounted on OMV," *Acta Astronautica*, Vol. 15, No. 12, 1987, pp. 981-986.

Umetani, Y. and Yoshida, K., "Resolved Motion Rate Control of Space Manipulators with Generalized Jacobian Matrix," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 3, June 1989, pp. 303-314.

Vafa, Z. and Dubowsky, S., "On The Dynamics of Manipulators in Space Using The Virtual Manipulator Approach," *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 1987, pp. 579-585.

Xu, Y., "The Measure of Dynamic Coupling of Space Robot Systems," *Proc. of IEEE Int. Conf. on Robotics and Automation*, Atlanta, Georgia, May 1993, pp. 615-620.

Yamada, K., Yoshikawa, S., and Fujita, Y., "Arm Path Planning of a Space Robot with Angular Momentum," *Journal of Advanced Robotics*, Vol. 9, No. 6, 1995, pp. 693-709.

Yokokohji, Y., Toyoshima, T., and Yoshikawa, T., "Efficient Computational Algorithms for Trajectory Control of Free-Flying Space Robots with Multiple Arms," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, October 1993, pp. 571-580.

Yoshida, K., Kurazume, R., and Umetani, Y., "Dual Arm Coordination in Space Free-Flying Robot," *Proc. of IEEE Int. Conf. on Robotics and Automation*, April 1991, pp. 2516-2521.

Yoshida, K. and Nenchev, D. N., "Space Robot Impact Analysis and Satellite-Base Impulse Minimization Using Reaction Null-Space," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nagoya, Japan, May 1995, pp. 1271-1277.

Yoshida, K., "Experimental Study on the Dynamics and Control of a Space Robot with Experimental Free-Floating Robot Satellite (EFFORTS) Simulators," *Journal of Advanced Robotics*, Vol. 9, No. 6, 1995, pp. 583-602.

Youcef-Toumi, K. and Ito, O., "Controller Design for Systems with Unknown Dynamics," *Proc. of American Cont. Conf.*, Minneapolis, USA, 1987.

Wee, L. B. and Walker, M. W., "On the Dynamics of Contact between Space Robots and Configuration Control for Impact Minimization," *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 5, October 1993, pp. 581-591.

Whitney, D. E., "Historical Perspective and State of the Art in Robot Force Control," *The International Journal of Robotics Research*, Vol. 6, No. 1, Spring 1987, pp. 3-14.

Appendix A

Three Formats Used in Dynamics Modelling

As discussed in Section 3.2.2.1, the system kinetic energy is composed of three typical terms which have to be differentiated according to Eq. (3.1). Differentiation of these terms, is presented in this appendix, to obtain three *formats* as used in deriving the system dynamics model.

Considering the first typical term, as given in Eq. (3.17a) and repeated here

$$a_1 = \frac{1}{2} m \, \dot{\mathbf{r}} \cdot \dot{\mathbf{r}} \tag{3.17a}$$

its differentiation with respect to $\dot{q}_i$ as an arbitrary generalized speed is obtained as

$$\frac{\partial a_1}{\partial \dot{q}_i} = m \frac{\partial \dot{\mathbf{r}}}{\partial \dot{q}_i} \cdot \dot{\mathbf{r}} \tag{A.1}$$

Note that for the implementation of the following formulation, $\mathbf{r}$ has to be differentiated in the inertial frame[13]. Then, $\dot{\mathbf{r}} = d\mathbf{r}/dt$ can be calculated as

---

13- If $\mathbf{r}$ is not differentiated in the inertial frame, then

$$\dot{\mathbf{r}} = {}^{B}\dot{\mathbf{r}} + \boldsymbol{\omega}_B \times \mathbf{r} \tag{A.1.a}$$

$$\dot{\mathbf{r}} = \sum_{s=1}^{N} \frac{\partial \mathbf{r}}{\partial q_s} \dot{q}_s \tag{A.2}$$

which yields

$$\frac{\partial \dot{\mathbf{r}}}{\partial \dot{q}_i} = \frac{\partial \mathbf{r}}{\partial q_i} \tag{A.3}$$

Substitution of Eq. (A.3) into Eq. (A.1) yields

$$\frac{\partial a_1}{\partial \dot{q}_i} = m \frac{\partial \mathbf{r}}{\partial q_i} \cdot \dot{\mathbf{r}} \tag{A.4}$$

which can be differentiated with respect to time, to obtain

$$\frac{d}{dt}\left(\frac{\partial a_1}{\partial \dot{q}_i}\right) = m \frac{\partial \dot{\mathbf{r}}}{\partial q_i} \cdot \dot{\mathbf{r}} + m \frac{\partial \mathbf{r}}{\partial q_i} \cdot \ddot{\mathbf{r}} \tag{A.5}$$

Also, a_1 can be differentiated with respect to q_i as an arbitrary generalized coordinate

$$\frac{\partial a_1}{\partial q_i} = m \frac{\partial \dot{\mathbf{r}}}{\partial q_i} \cdot \dot{\mathbf{r}} \tag{A.6}$$

Therefore, based on Eqs. (A.5) and (A.6), it can be written

$$\frac{d}{dt}\left(\frac{\partial a_1}{\partial \dot{q}_i}\right) - \frac{\partial a_1}{\partial q_i} = m \frac{\partial \mathbf{r}}{\partial q_i} \cdot \ddot{\mathbf{r}} \tag{A.7}$$

where $\ddot{\mathbf{r}}$ can be obtained as

---

where ${}^{B}\dot{\mathbf{r}}$ is time differentiation of $\mathbf{r}$ when expressed in a frame B which has an angular velocity of $\boldsymbol{\omega}_B$ with respect to the inertial frame, and can be computed as

$$^{B}\dot{\mathbf{r}} = \sum_{s=1}^{N} \frac{^{B}\partial \mathbf{r}}{\partial q_s} \dot{q}_s + \frac{^{B}\partial \mathbf{r}}{\partial t} \tag{A.1.b}$$

Note that a left superscript on partial derivatives denotes the frame in which the differentiation has to be taken. Therefore, unless ${}^{B}\partial \mathbf{r}/\partial t = 0$, it can be seen that

$$\frac{^{B}\partial \dot{\mathbf{r}}}{\partial \dot{q}_s} \neq \frac{^{B}\partial \mathbf{r}}{\partial q_s} \tag{A.1.c}$$

which necessitates the condition of differentiating $\mathbf{r}$ in the inertial frame, for writing Eq. (A.2-3).

$$\ddot{\mathbf{r}} = \sum_{s=1}^{N} \left\{ \frac{\partial}{\partial q_s} \left(\sum_{t=1}^{N} \frac{\partial \mathbf{r}}{\partial q_t} \dot{q}_t \right) \dot{q}_s + \frac{\partial \mathbf{r}}{\partial q_s} \ddot{q}_s \right\} \tag{A.8}$$

Substitution of Eq. (A.8) into Eq. (A.7), and further simplifications, yield

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial a_1}{\partial \dot{q}_i}\right) - \frac{\partial a_1}{\partial q_i} = \left[m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\frac{\partial \mathbf{r}}{\partial q_1} \quad \cdots \quad m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\frac{\partial \mathbf{r}}{\partial q_N} \right] \ddot{\mathbf{q}} + \\ \left[m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}}{\partial q_s \partial q_1}\dot{q}_s\right) \quad \cdots \quad m\frac{\partial \mathbf{r}}{\partial q_i}\cdot\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}}{\partial q_s \partial q_N}\dot{q}_s\right) \right] \dot{\mathbf{q}} \end{aligned} \tag{A.9}$$

which describes *format-I*, given as Eq. (3.18), where $\mathbf{r}$ has to be differentiated in the inertial frame.

Next, considering the second term, as given in Eq. (3.17b)

$$a_2 = \frac{1}{2}\boldsymbol{\omega}\cdot\mathbf{I}\cdot\boldsymbol{\omega} \tag{3.17b}$$

its differentiation with respect to q_i as an arbitrary generalized coordinate, is

$$\frac{\partial a_2}{\partial q_i} = \boldsymbol{\omega}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial q_i} \tag{A.10}$$

where $\boldsymbol{\omega}$ is differentiated in the body frame. Also, differentiation of a_2 with respect to $\dot{q}_i$ as an arbitrary generalized speed, is obtained as[14]

$$\frac{\partial a_2}{\partial \dot{q}_i} = \boldsymbol{\omega}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i} \tag{A.11}$$

which can be differentiated with respect to time, to obtain

$$\frac{d}{dt}\left(\frac{\partial a_2}{\partial \dot{q}_i}\right) = \dot{\boldsymbol{\omega}}\cdot\mathbf{I}\cdot\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i} + \boldsymbol{\omega}\cdot\mathbf{I}\cdot\frac{d}{dt}\left(\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i}\right) \tag{A.12}$$

Then, $\dot{\boldsymbol{\omega}}$ can be computed as

14- Considering Eq. (A.1.a), since $\boldsymbol{\omega}\times\boldsymbol{\omega} = \mathbf{0}$, the time derivative of a body's angular velocity in both inertial and corresponding body frame is the same. Therefore, it is preferable to implement all differentiations related to a_1 in an appropriate body frame. Hence, the angular velocity of an individual body ($\boldsymbol{\omega}$) is differentiated in the corresponding body frame, where $\mathbf{I}$ is a constant.

$$\dot{\boldsymbol{\omega}} = \sum_{s=1}^{N} \left\{ \frac{\partial \boldsymbol{\omega}}{\partial q_s} \dot{q}_s + \frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_s} \ddot{q}_s \right\} \tag{A.13}$$

Also

$$\frac{d}{dt}\left(\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i}\right) = \sum_{s=1}^{N} \left\{ \frac{\partial^2 \boldsymbol{\omega}}{\partial \dot{q}_i \partial q_s} \dot{q}_s + \frac{\partial^2 \boldsymbol{\omega}}{\partial \dot{q}_i \partial \dot{q}_s} \ddot{q}_s \right\} = \sum_{s=1}^{N} \frac{\partial^2 \boldsymbol{\omega}}{\partial \dot{q}_i \partial q_s} \dot{q}_s \tag{A.14}$$

Note that the angular velocity of an individual body is a linear function of generalized speeds, see Eqs. (2.12) and (3.7), therefore $\partial^2 \boldsymbol{\omega} / \partial \dot{q}_i \partial \dot{q}_s = \mathbf{0}$ in Eq. (A.14).

Substitution of Eqs. (A.13) and (A.14) into Eq. (A.12), and subtract Eq. (A.10) from the result, after further simplifications yield

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial a_2}{\partial \dot{q}_i}\right) - \frac{\partial a_2}{\partial q_i} = &\left[\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i} \cdot \mathbf{I} \cdot \frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_1} \quad \cdots \quad \frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i} \cdot \mathbf{I} \cdot \frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_N} \right] \ddot{\mathbf{q}} + \\ &\left[\frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i} \cdot \mathbf{I} \cdot \frac{\partial \boldsymbol{\omega}}{\partial q_1} + \boldsymbol{\omega} \cdot \mathbf{I} \cdot \frac{\partial^2 \boldsymbol{\omega}}{\partial \dot{q}_i \partial q_1} \quad \cdots \quad \frac{\partial \boldsymbol{\omega}}{\partial \dot{q}_i} \cdot \mathbf{I} \cdot \frac{\partial \boldsymbol{\omega}}{\partial q_N} + \boldsymbol{\omega} \cdot \mathbf{I} \cdot \frac{\partial^2 \boldsymbol{\omega}}{\partial \dot{q}_i \partial q_N} \right] \dot{\mathbf{q}} - \boldsymbol{\omega} \cdot \mathbf{I} \cdot \frac{\partial \boldsymbol{\omega}}{\partial q_i} \end{aligned} \tag{A.15}$$

which describes *format-II*, given as Eq. (3.19), and can be considered as contribution of the second term to the equations of motion. Note that $\boldsymbol{\omega}$ is differentiated in a body frame in which $\mathbf{I}$ is considered as a constant dyad.

Finally, considering the third typical term of the system kinetic energy, as defined in Eq. (3.17c)

$$a_3 = \dot{\mathbf{R}}_{C_0} \cdot \sum_k m_k \dot{\mathbf{r}}_k \tag{3.17c}$$

its differentiation with respect to q_i is

$$\frac{\partial a_3}{\partial q_i} = \frac{\partial \dot{\mathbf{R}}_{C_0}}{\partial q_i} \cdot \left(\sum_k m_k \dot{\mathbf{r}}_k \right) + \dot{\mathbf{R}}_{C_0} \cdot \left(\sum_k m_k \frac{\partial \dot{\mathbf{r}}_k}{\partial q_i} \right) \tag{A.16}$$

and its differentiation with respect to $\dot{q}_i$, can be obtained as

$$\frac{\partial a_3}{\partial \dot{q}_i} = \frac{\partial \mathbf{R}_{C_0}}{\partial q_i} \cdot \left(\sum_k m_k \dot{\mathbf{r}}_k \right) + \dot{\mathbf{R}}_{C_0} \cdot \left(\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_i} \right) \tag{A.17}$$

where all derivatives are computed in the inertial frame. Then, Eq. (A.17) can be differentiated with respect to time, which yields

$$\frac{d}{dt}\left(\frac{\partial a_3}{\partial \dot{q}_i}\right)=\frac{\partial \dot{\mathbf{R}}_{C_0}}{\partial q_i}\cdot\left(\sum_k m_k \dot{\mathbf{r}}_k\right)+\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\left(\sum_k m_k \ddot{\mathbf{r}}_k\right)+ \\ \ddot{\mathbf{R}}_{C_0}\cdot\left(\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_i}\right)+\dot{\mathbf{R}}_{C_0}\cdot\left(\sum_k m_k \frac{\partial \dot{\mathbf{r}}_k}{\partial q_i}\right) \tag{A.18}$$

Therefore, subtracting Eq. (A.16) from Eq. (A.18) yields

$$\frac{d}{dt}\left(\frac{\partial a_3}{\partial \dot{q}_i}\right)-\frac{\partial a_3}{\partial q_i}=\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\left(\sum_k m_k \ddot{\mathbf{r}}_k\right)+\ddot{\mathbf{R}}_{C_0}\cdot\left(\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_i}\right) \tag{A.19}$$

where $\ddot{\mathbf{r}}_k$ and $\ddot{\mathbf{R}}_{C_0}$ can be written in terms of generalized coordinates and their rates as given in Eq. (A.8). Substitution of these vectors by appropriate expressions, and further simplifications, leads to

$$\frac{d}{dt}\left(\frac{\partial a_3}{\partial \dot{q}_i}\right)-\frac{\partial a_3}{\partial q_i}=\left[\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_1} \quad \cdots \quad \frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_N}\right]\ddot{\mathbf{q}}+ \\ \left[\frac{\partial \mathbf{R}_{C_0}}{\partial q_1}\cdot\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_i} \quad \cdots \quad \frac{\partial \mathbf{R}_{C_0}}{\partial q_N}\cdot\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_i}\right]\ddot{\mathbf{q}}+ \\ \left[\frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}_k}{\partial q_1 \partial q_s}\dot{q}_s\right) \quad \cdots \quad \frac{\partial \mathbf{R}_{C_0}}{\partial q_i}\cdot\sum_k m_k\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{r}_k}{\partial q_N \partial q_s}\dot{q}_s\right)\right]\dot{\mathbf{q}}+ \\ \left[\left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{R}_{C_0}}{\partial q_1 \partial q_s}\dot{q}_s\right)\cdot\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_i} \quad \cdots \quad \left(\sum_{s=1}^{N}\frac{\partial^2 \mathbf{R}_{C_0}}{\partial q_N \partial q_s}\dot{q}_s\right)\cdot\sum_k m_k \frac{\partial \mathbf{r}_k}{\partial q_i}\right]\dot{\mathbf{q}} \tag{A.20}$$

which describes *format-III*, given as Eq. (3.20), where $\mathbf{R}_{C_0}$ and $\mathbf{r}_k$ have to be differentiated in the inertial frame.

Appendix B

Case Study: Transfer Functions

Considering the simple model depicted in Figure 6.2, the corresponding transfer functions for the MIC and OIC algorithms are presented here. The open-loop block diagram for this system, based on the Laplace transformation of Eqs. (6.31), is shown in Figure B.1. Next, the MIC and OIC laws as applied to the considered system, are derived.

Considering Eqs. (6.19) and (6.22), the MIC algorithm yields the following control force as applied to this system

$$F_1 = f_m + f_f \tag{B.1}$$

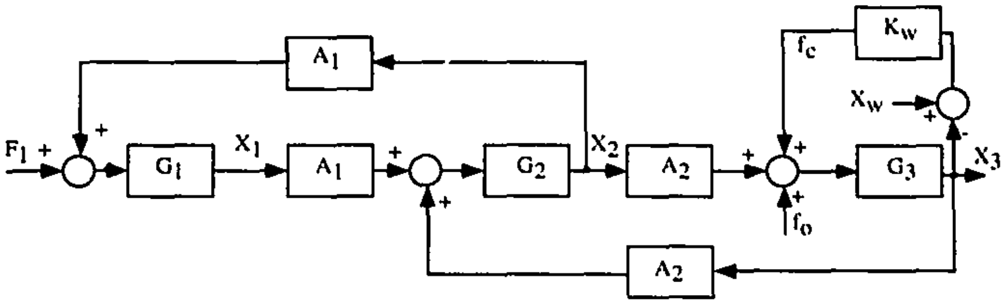
where

$$\begin{aligned} f_m &= m_1\, m_{des}^{-1}(m_{des}\ddot{x}_{1des} + k_d\dot{e}_1 + k_p e_1 + f_c) + b_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2 + l_{1free}) \\ &\quad + m_2\, m_{des}^{-1}(m_{des}\ddot{x}_{2des} + k_d\dot{e}_2 + k_p e_2 + f_c) + b_1(\dot{x}_2 - \dot{x}_1) + b_2(\dot{x}_2 - \dot{x}_3) \\ &\quad + k_1(x_2 - x_1 - l_{1free}) + k_2(x_2 - x_3 + l_{2free}) \end{aligned} \tag{B.2a}$$

$$f_f = m_3\, m_{des}^{-1}(m_{des}\ddot{x}_{3des} + k_d\dot{e}_3 + k_p e_3 + f_c) + b_2(\dot{x}_3 - \dot{x}_2) + k_2(x_3 - x_2 - l_{2free}) - (f_o + f_c) \tag{B.2b}$$

assuming that the exact value of the contact force, f_c, is available as

$$f_c = k_w\,(x_w - x_3) \tag{B.2c}$$



$$G_1 = \frac{1}{m_1 s^2 + b_1 s + K_1} \qquad G_2 = \frac{1}{m_2 s^2 + (b_1 + b_2)s + K_1 + K_2} \qquad G_3 = \frac{1}{m_3 s^2 + b_2 s + K_2}$$

$$A_1 = b_1 s + K_1 \qquad A_2 = b_2 s + K_2$$

Figure B.1: Block diagram for the open-loop system.

where k_w is stiffness coefficient of the obstacle located at x_w. Note that the desired trajectories for m_1, and m_2 can be defined based on the desired trajectory for the object (m_3), as

$$\begin{aligned} x_{2_{des}} &= x_{3_{des}} - l_{2_{free}} \\ x_{1_{des}} &= x_{3_{des}} - l_{1_{free}} - l_{2_{free}} \end{aligned} \tag{B.3}$$

Substituting Eq. (B.1) into (6.31), and summing the result, yields

$$\begin{aligned} m_1\,(m_{des}\ddot{e}_1 + k_d\dot{e}_1 + k_p e_1 + f_c) + m_2\,(m_{des}\ddot{e}_2 + k_d\dot{e}_2 + k_p e_2 + f_c) + \\ m_3\,(m_{des}\ddot{e}_3 + k_d\dot{e}_3 + k_p e_3 + f_c) = 0 \end{aligned} \tag{B.4}$$

Since Eq. (B.4) must hold for any set of m_1, m_2, and m_3, it can be concluded that

$$\begin{aligned} m_{des}\ddot{e}_1 + k_d\dot{e}_1 + k_p e_1 + f_c &= 0 \\ m_{des}\ddot{e}_2 + k_d\dot{e}_2 + k_p e_2 + f_c &= 0 \\ m_{des}\ddot{e}_3 + k_d\dot{e}_3 + k_p e_3 + f_c &= 0 \end{aligned} \tag{B.5}$$

which reveals that all tracking errors are governed by the same target impedance.

The OIC as applied to the considered system, yields the following control force[15]

$$F_1 = f_{cmp} + f_{cmd} \tag{B.6}$$

15- For details, see Schneider and Cannon (1992).

where

$$f_{cmp} = m_1 \ddot{x}_{cmd} + b_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2 + l_{1_{free}})$$
$$+ m_2 \ddot{x}_{cmd} + b_1(\dot{x}_2 - \dot{x}_1) + b_2(\dot{x}_2 - \dot{x}_3) + k_1(x_2 - x_1 - l_{1_{free}}) + k_2(x_2 - x_3 + l_{2_{free}}) \quad \text{(B.7a)}$$

$$f_{cmd} = m_3 \ddot{x}_{cmd} + b_2(\dot{x}_3 - \dot{x}_2) + k_2(x_3 - x_2 - l_{2_{free}}) - (f_o + f_c) \quad \text{(B.7b)}$$

and

$$\ddot{x}_{cmd} = m_{des}^{-1}(m_{des}\ddot{x}_{3_{des}} + k_d\dot{e}_3 + k_p e_3 + f_c) \quad \text{(B.7c)}$$

To obtain the transfer function between the output, i.e. object position, and the given desired position, corresponding block diagrams for the two algorithm are simplified, Figures B.2 and B.3. Note that to obtain a deeper insight of the nature of these algorithms, mass properties in the controller circuit are considered different from the corresponding true parameters. Therefore, m_i represents true mass value which appears in G_i, while $\hat{m}_i$ is the given value for control purposes. For root locus analysis, the object stiffness coefficient k_2 was selected as a variable parameter. So, the characteristic equation for the corresponding transfer functions, $G_{MIC}(s)$ and $G_{OIC}(s)$, can be written as

$$1 + k_2 \frac{N(s)}{D(s)} = 0 \quad \text{(B.8)}$$

In the following, $G_{MIC}(s)$ and $G_{OIC}(s)$ are presented in a proper format to yield the corresponding characteristic equation in the given form.

❒ **For the MIC:**

$$G_{MIC} = \frac{x_3}{x_{des_3}} = \frac{Num_1}{Den_1} \quad \text{(B.9)}$$

where

$$Num_1 = (m_{des}s^2 + k_d s + k_p)(\hat{m}_1 + \hat{m}_2 + \hat{m}_3)(b_1 s + k_1)(b_2 s + k_2) \quad \text{(B.10a)}$$

$$Den_1 = D_1(s) + k_2 N_1(s) \quad \text{(B.10b)}$$

where

$$
\begin{aligned}
N_1(s) = {} & m_{des}(m_2+m_3)m_1 s^4 + \left(\hat{m}_1(m_2+m_3)k_d + m_{des}(m_1+m_2+m_3)b_1\right)s^3 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_1k_d + m_{des}(m_1+m_2+m_3)k_1 + m_{des}m_1k_w\right)s^2 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_1k_p + (\hat{m}_1+\hat{m}_2+\hat{m}_3)k_1k_d + (m_1+m_2)b_1k_w + \hat{m}_1k_wk_d\right)s + \\
& (\hat{m}_1+\hat{m}_2+\hat{m}_3)k_1k_p + \hat{m}_1k_wk_p + (m_1+m_2+m_3)k_wk_1
\end{aligned}
\tag{B.11a}
$$

$$
\begin{aligned}
D_1(s) = {} & m_{des}m_1m_2m_3s^6 + \left(\hat{m}_1m_2m_3k_d + m_{des}(m_1+m_2)m_3b_1 + m_{des}(m_2+m_3)m_1b_2\right)s^5 + \\
& \left(m_{des}(m_1+m_2)m_3k_1 + m_{des}(m_1+m_2+m_3)b_1b_2 + \hat{m}_2m_3b_1k_d + \hat{m}_1(m_2b_2+m_3(b_1+b_2))k_d + \right. \\
& \left. \hat{m}_1m_2m_3k_p + m_{des}m_1m_2k_w\right)s^4 + \left(\hat{m}_1(m_2+m_3)b_2k_p + m_3(\hat{m}_1+\hat{m}_2)(b_1k_p+k_1k_d) + \right. \\
& (\hat{m}_1+\hat{m}_2+\hat{m}_3)b_1b_2k_d + m_{des}(m_1+m_2+m_3)k_1b_2 + m_{des}m_1(b_1+b_2)k_w + \\
& \left. (m_{des}b_1+\hat{m}_1k_p)m_2k_w\right)s^3 + \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_2(b_1k_p+k_1k_d) + m_3(\hat{m}_1+\hat{m}_2)k_1k_p + \right. \\
& \left. (\hat{m}_1+\hat{m}_2)b_1k_dk_w + m_{des}(m_1+m_2)k_1k_w + (m_1+m_2+m_3)b_1b_2k_w + \hat{m}_1(b_2k_d+m_2k_p)k_w\right)s^2 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_2k_1k_p + (\hat{m}_1+\hat{m}_2)k_1k_dk_w + \hat{m}_1(b_1+b_2)k_pk_w + (m_1+m_2+m_3)k_1b_2k_w + \right. \\
& \left. \hat{m}_2k_pb_1k_w\right)s + (\hat{m}_1+\hat{m}_2)k_1k_pk_w
\end{aligned}
\tag{B.11b}
$$

❒ **For the OIC:**

$$
G_{OIC}(s) = \frac{x_3}{x_{des_3}} = \frac{Num_2}{Den_2} \tag{B.12}
$$

where

$$
Num_2 = (m_{des}s^2 + k_ds + k_p)(\hat{m}_1+\hat{m}_2+\hat{m}_3)(b_1s+k_1)(b_2s+k_2) \tag{B.13a}
$$

$$
Den_2 = D_2(s) + k_2N_2(s) \tag{B.13b}
$$

where

$$
\begin{aligned}
N_2(s) = {} & m_{des}(m_2+m_3)m_1s^4 + m_{des}(m_1+m_2+m_3)b_1s^3 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_1k_d + m_{des}(m_1+m_2+m_3)k_1 + m_{des}m_1k_w\right)s^2 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_1k_p + (\hat{m}_1+\hat{m}_2+\hat{m}_3)k_1k_d + (m_1+m_2+m_3)b_1k_w\right)s + \\
& (\hat{m}_1+\hat{m}_2+\hat{m}_3)k_1k_p + (m_1+m_2+m_3)k_wk_1
\end{aligned}
\tag{B.14a}
$$

$$
\begin{aligned}
D_2(s) = {} & m_{des}m_1m_2m_3s^6 + \left(m_{des}(m_1+m_2)m_3b_1 + m_{des}(m_2+m_3)m_1b_2\right)s^5 + \\
& \left(m_{des}(m_1+m_2)m_3k_1 + m_{des}(m_1+m_2+m_3)b_1b_2 + m_{des}m_1m_2k_w\right)s^4 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_1b_2k_d + m_{des}(m_1+m_2+m_3)k_1b_2 + m_{des}m_1(b_1+b_2)k_w + m_{des}b_1m_2k_w\right)s^3 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_2(b_1k_p+k_1k_d) + (m_1+m_2+m_3)b_1b_2k_w + m_{des}(m_1+m_2)k_1k_w\right)s^2 + \\
& \left((\hat{m}_1+\hat{m}_2+\hat{m}_3)b_2k_1k_p + (m_1+m_2+m_3)k_1b_2k_w\right)s
\end{aligned}
\tag{B.14b}
$$

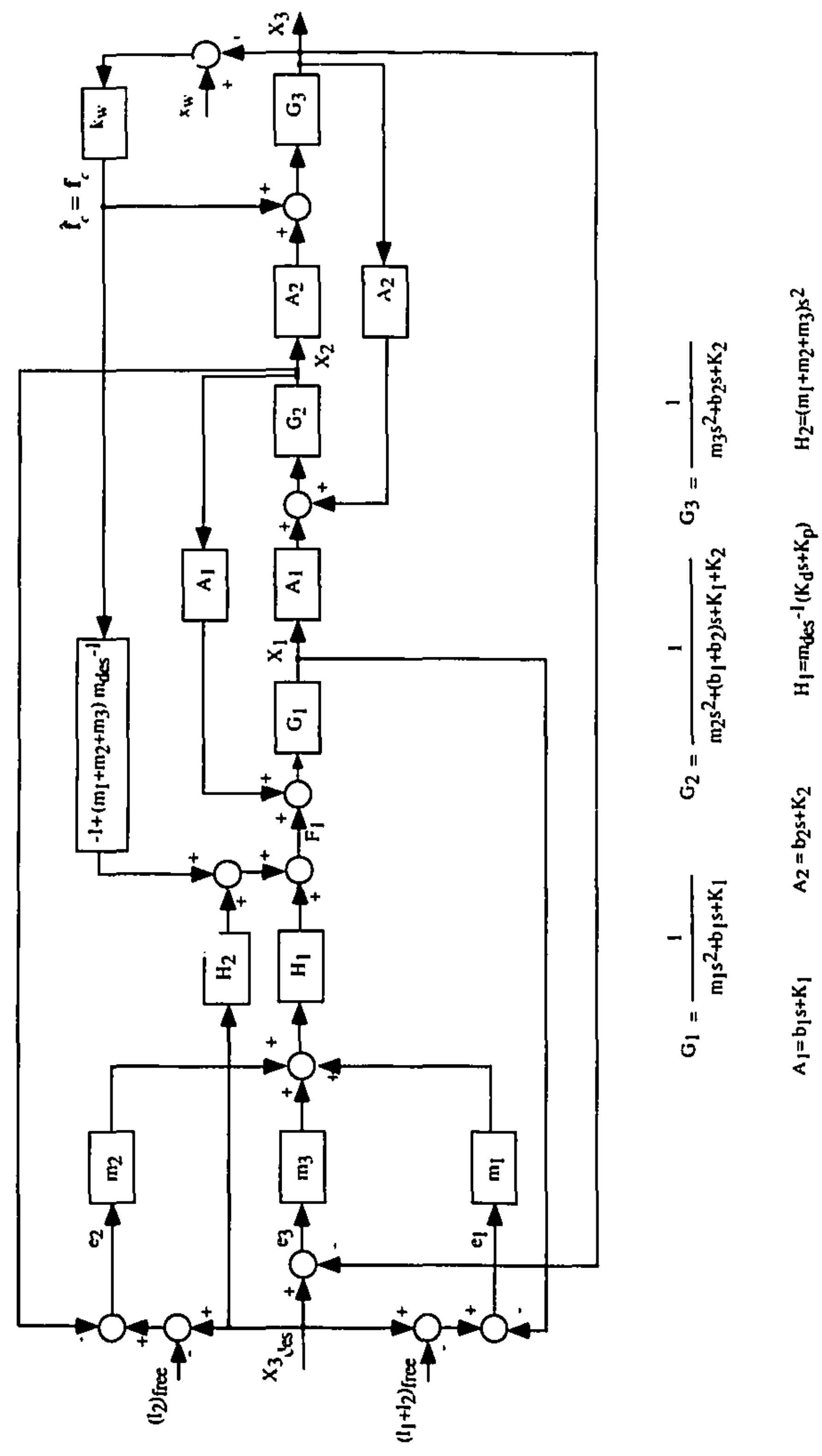
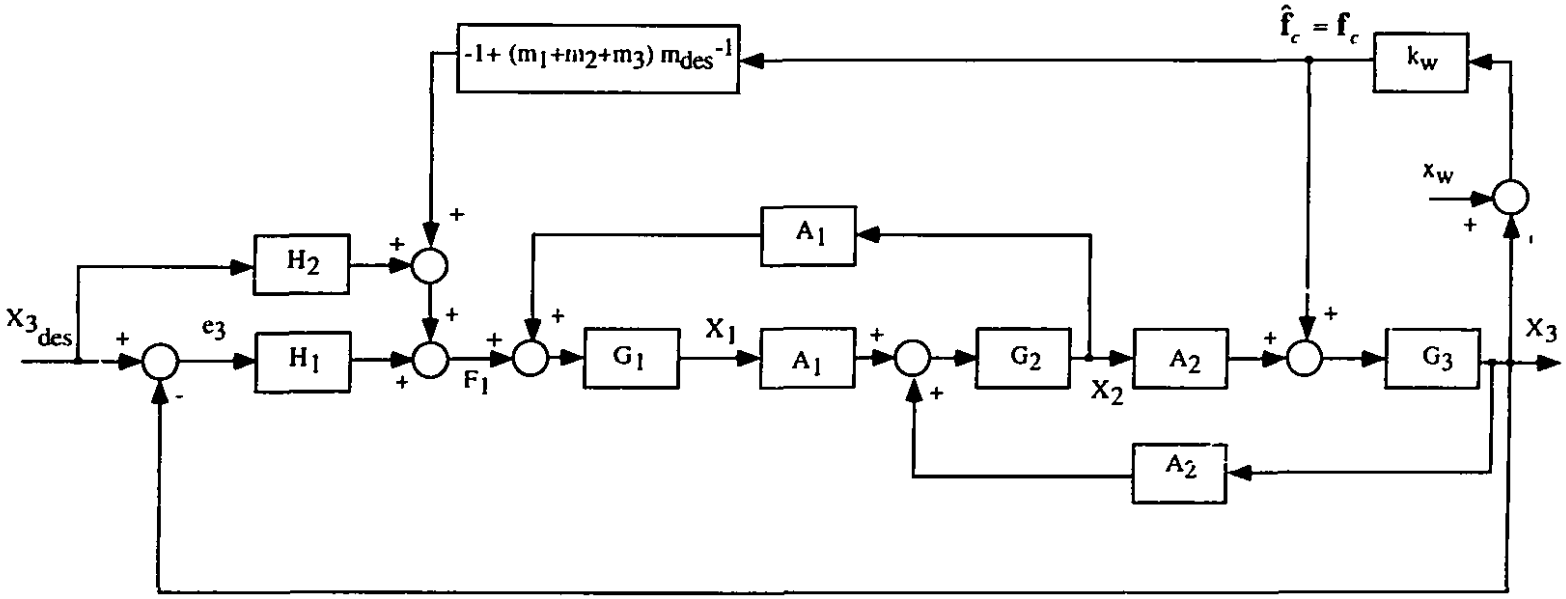


Figure B.2: Block diagram for the MIC implementation.



$$G_1 = \frac{1}{m_1 s^2 + b_1 s + K_1} \qquad G_2 = \frac{1}{m_2 s^2 + (b_1 + b_2)s + K_1 + K_2} \qquad G_3 = \frac{1}{m_3 s^2 + b_2 s + K_2}$$

$$H_1 = (m_1 + m_2 + m_3) m_{des}^{-1} (K_d s + K_p) \qquad H_2 = (m_1 + m_2 + m_3) s^2 \qquad A_1 = b_1 s + K_1 \qquad A_2 = b_2 s + K_2$$

Figure B.3: Block diagram for the OIC implementation.

Appendix C

Case Study: Root Locus Analysis

Considering the system depicted in Figure 6.2, a root locus analysis for the MIC and OIC algorithms, is presented in this appendix. To this end, the loci are plotted as a function of the object stiffness (k_2) for various damping factors (b_2).

The system mass parameters are chosen as $m_1 = 100\,kg$, $m_2 = 20.0\,kg$, and $m_3 = 10.0\,kg$. Assuming a fundamental frequency of 20 Hz for the manipulator (which is relatively high, according to Rivin (1988)), k_1 is computed as

$$\omega = 2\pi f = \sqrt{\frac{k_1(m_1 + m_2)}{m_1 m_2}} \quad \Rightarrow k_1 = 2.6 \times 10^5\ N/m$$

Also, considering a logarithmic decrement (δ) of 0.2 for the manipulator (which is again a relatively large structural damping, according to Rivin, 1988), b_1 is computed as

$$\zeta = \frac{\delta}{2\pi} = 0.03 \Rightarrow b_1 = 2\zeta\sqrt{k_1 m_1} = 325\ kg/\sec$$

Unless otherwise stated, the controller parameters are $m_{des} = 100.0$, $k_p = 100.0$, $k_d = 300.0$, $\hat{m}_1 = 110\,kg$, $\hat{m}_2 = 18\,kg$, and $\hat{m}_3 = 11\,kg$. The variable parameter k_2, is changing between 0 and 10^{10}. For the obstacle, see Figure 6.2, k_w is equal to 10^5 if contact occurs, otherwise it is zero.

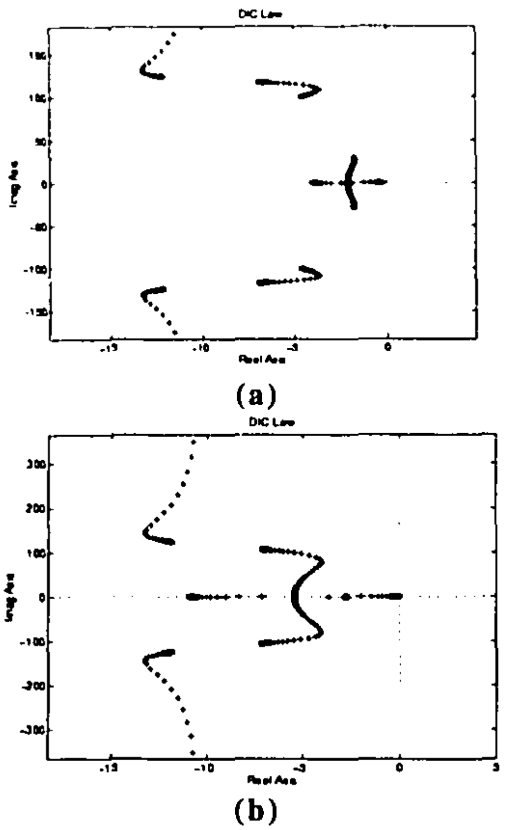


Figure C.1: Root locus for the MIC law, $b_2 = 100$, (a) In contact (b) No contact.

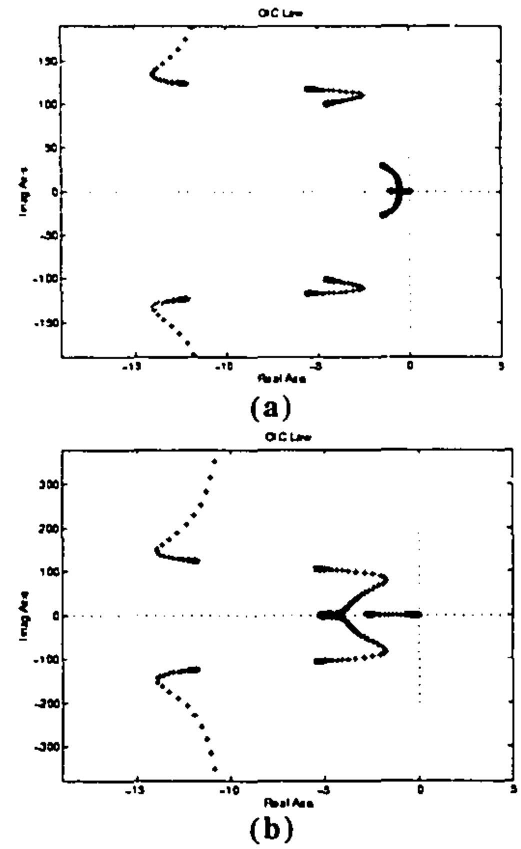
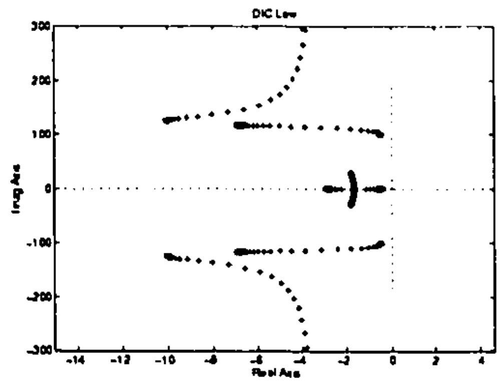


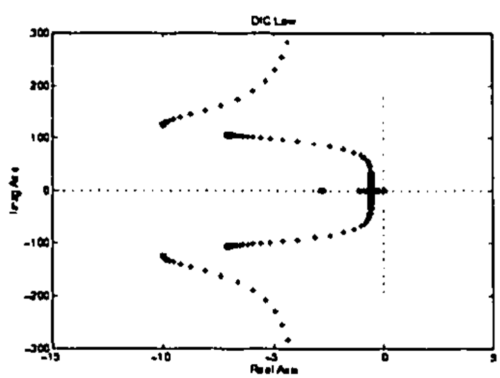
Figure C.2: Root locus for the OIC law, $b_2 = 100$, (a) In contact (b) No contact.

Figures C.1,2 compare the root loci of the MIC and OIC, for $b_2 = 100.0 kg/\text{sec}$. As it is seen, both algorithms are stable, no matter whether the object is in contact with the obstacle or not. Figures C.3, and C.4 compare these root loci, for $b_2 = 10.0 kg/\text{sec}$. Here, it can be seen that both algorithms are stable if the object is in contact with the obstacle, but the OIC becomes unstable if there is no contact. Note that contact between the object and obstacle, adds a kind of feedback to the system, and so results in different behavior. Next, we see the effect of different controller parameters on the stability of the OIC algorithm.

The effect of choosing larger gains and the desired mass parameter on the stability of OIC algorithm, for $b_2 = 10.0 kg/\text{sec}$ with no contact, is shown in Figure C.5. In part (a), $m_{des} = 100.0$, $k_p = 1000.0$, and $k_d = 300.0$, while in part (b), $m_{des} = 100.0$, $k_p = 100.0$, and $k_d = 1000.0$. As it is seen, choosing larger gains does not result in a stable system. In part (c), $m_{des} = 500.0$, $k_p = 100.0$, and $k_d = 300.0$, while in part (d), $m_{des} = 500.0$, $k_p = 100.0$, and $k_d = 1000.0$. A larger value of the desired mass has a positive effect on the stability of this algorithm, as can be seen in Figure C.5c. However, it is expected (and will be shown by simulation) that selecting a higher inertia for the desired object impedance results in a sluggish performance. Choosing a larger k_d besides larger value for the desired mass results in a more stable root locus, Figure C.5d.

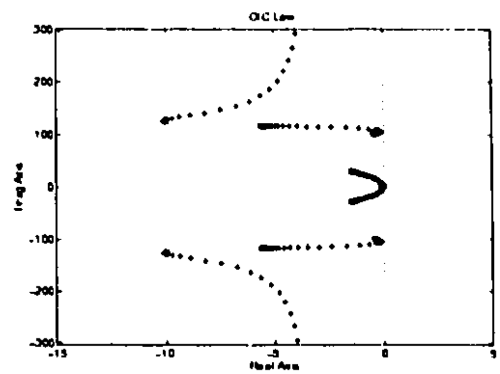


(a)

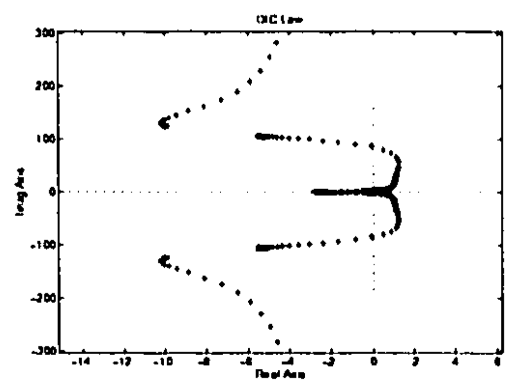


(b)

Figure C.3: Root locus for the MIC law, $b_2 = 10$, (a) In contact (b) No contact.



(a)



(b)

Figure C.4: Root locus for the OIC law, $b_2 = 10$, (a) In contact (b) No contact.

Figures C.6 and C.7 compare the root locus of the MIC and OIC algorithms, for $b_2 = 0.0$. As it is seen, the MIC algorithm is stable no matter whether the object is in contact with the obstacle or not. On the contrary, the OIC algorithm is unstable and, for this case, choosing a larger desired mass parameter or a larger k_d does not result in a stable root locus.

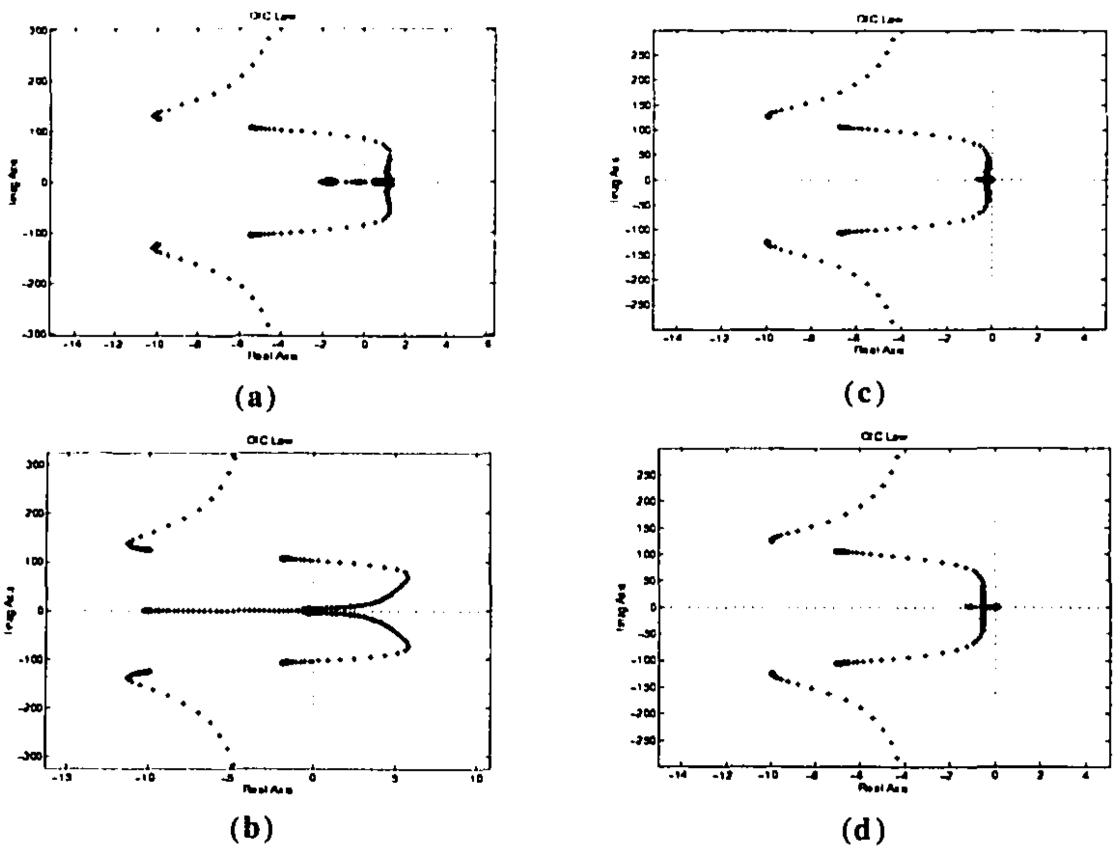
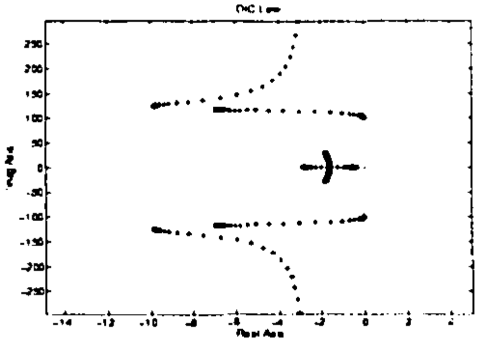
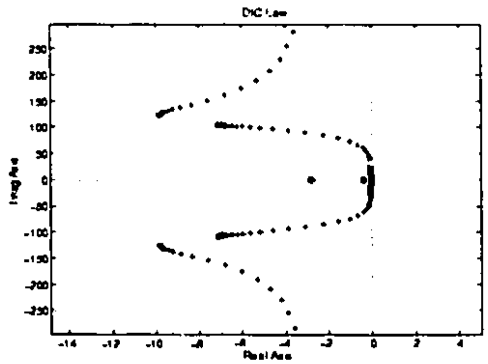


Figure C.5: Root locus for the OIC law, $b_2 = 10$, and no contact, (a) Larger K_p, (b) Larger K_d, (c) Larger m_{des}, (d) Larger K_d and m_{des}.

As shown in this appendix, the MIC algorithm has superior stability properties compared to OIC.

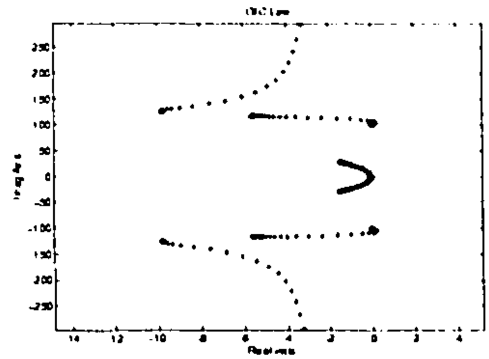


(a)

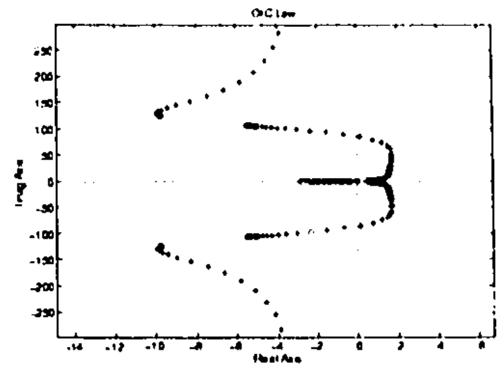


(b)

Figure C.6: Root locus for the MIC law, $b_2 = 0.0$, (a) In contact (b) No contact.



(a)



(b)

Figure C.7: Root locus for the OIC law, $b_2 = 0.0$, (a) In contact (b) No contact.

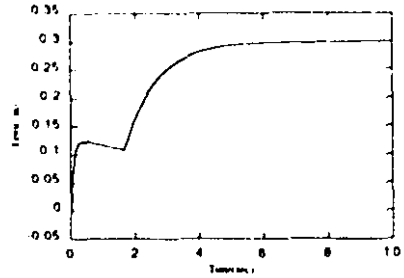
Appendix D

Case Study: Simulation Results

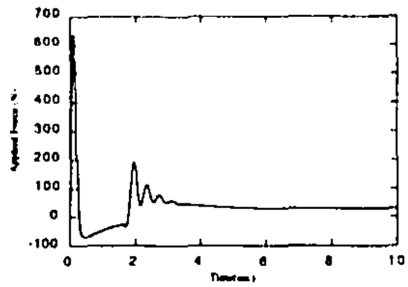
Considering the system depicted in Figure 6.2, and the set of controller parameters given in Section 6.3.4.4, it was shown by simulation that the system never rests under the OIC law, while the MIC algorithm yields a smooth stop of the object at the obstacle. Further investigations on the effect of choosing larger gains or the desired mass parameter, are presented in this appendix.

Choosing a larger damping gain[16], k_d=700, the obtained results are depicted in Figures D.1 and D.2. As it is seen, the resulting oscillations in applying the OIC law do not disappear, while the MIC algorithm yields a well-damped smoother response. Comparing these results to the previous ones (depicted in Figures 6.4,5), tracking errors in free motion (before the contact at $t \approx 2.0$ sec) are reduced as expected, and the peak of the input force increases for both algorithms. However, the contact force (particularly for the first impact) has an increase of about 30% for the OIC, while the MIC does not result in a substantial increase. Note that the root locus analysis shows that both the OIC and MIC are stable for both the "no contact" and "in contact" phases. However, choosing larger k_d's than this will make the system under the OIC law unstable.

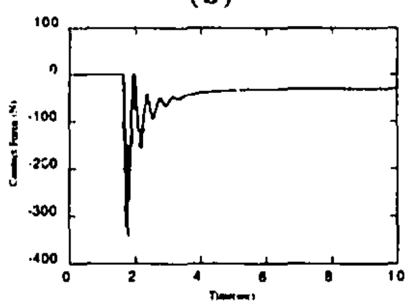
16- The controller parameters were formerly chosen as m_{des}=100, k_p=100, and k_d=300, see Section 6.3.4.4.



(a)

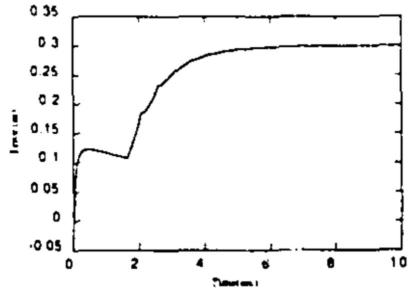


(b)

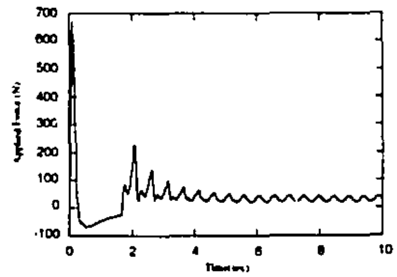


(c)

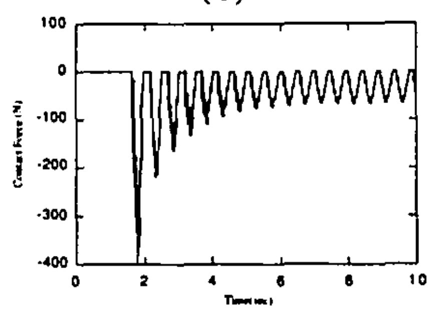
Figure D.1: Performance of the MIC, k_d =700, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.



(a)



(b)



(c)

Figure D.2: Performance of the OIC, k_d =700, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.

To see the effect of actuator saturation limits on the performance of the two algorithms, the previous simulation with k_d=700, is repeated for $-80 \le F_i \le 80$, Figures D.3 and D.4. Initially, both algorithms demand an input force which is beyond the saturation limit. Therefore, tracking errors grow, and both algorithms yield almost the same result. However, like previous cases, a big difference between the performance of the two algorithms appears when the demand is below the actuator saturation limit, particularly after the contact (at $t \approx 2.0$ sec). The MIC algorithm results in a smooth stop of the object at the obstacle, Figure D.3, while the system enters a limit cycle under the OIC law, Figure D.4.

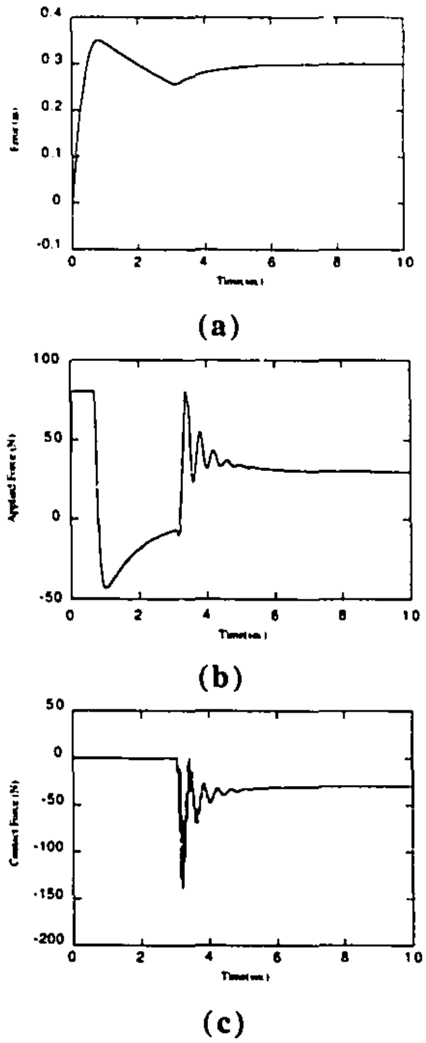


Figure D.3: Performance of the MIC with actuator saturation limit, $|F_{actuator}| < 80$, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.

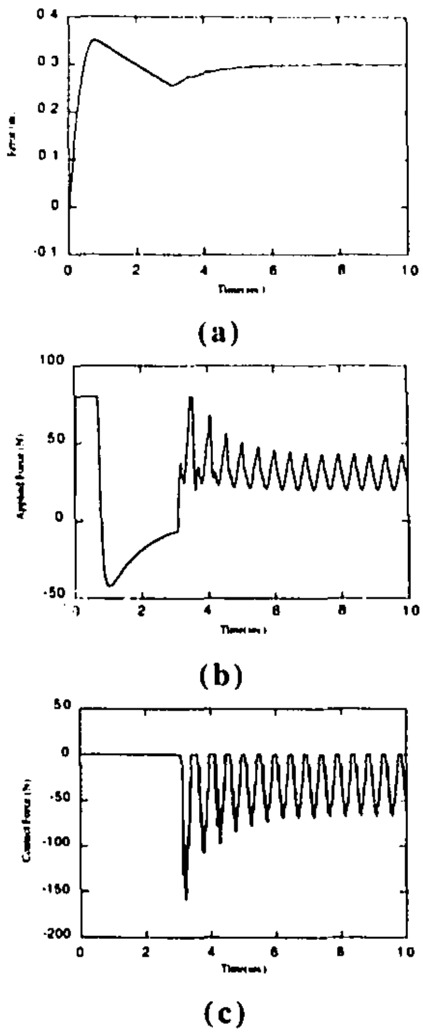
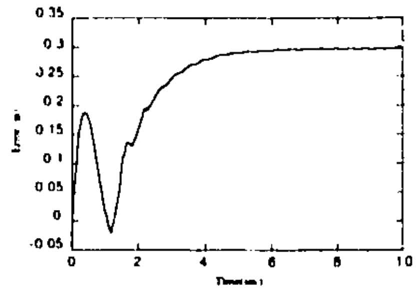
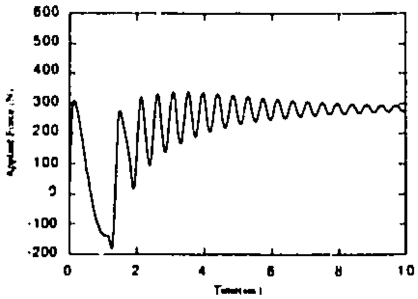


Figure D.4: Performance of the OIC with actuator saturation limit, $|F_{actuator}| < 80$, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.

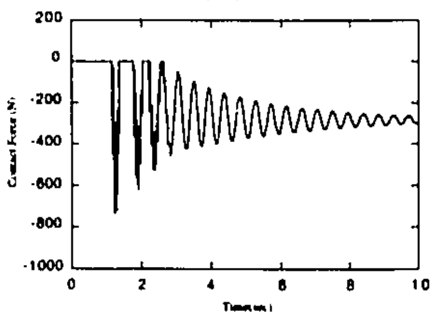
Figures D.5 and D.6 compare the result of choosing k_p=1000. As it is seen, the amplitude of the oscillations increase, when applying the OIC law, while the MIC algorithm yields a lower damped response (compared to k_p=100) which is expected. Note that root locus analysis shows that both OIC and MIC [illegible]e stable in both phases, but it seems so that OIC becomes unstable. In general, an on-off type of nonlinear system may become unstable or experience a limit cycle while it is switching between two linear stable systems. Longer simulation times show that the OIC is just experiencing a limit cycle, like previous cases.



(a)

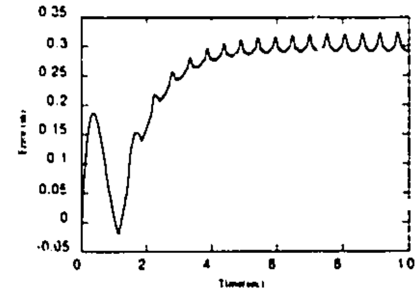


(b)

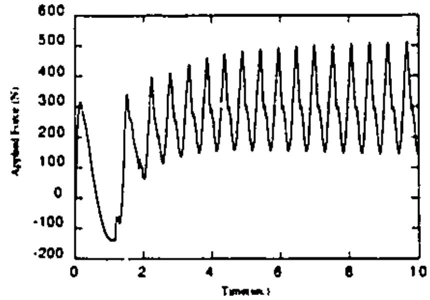


(c)

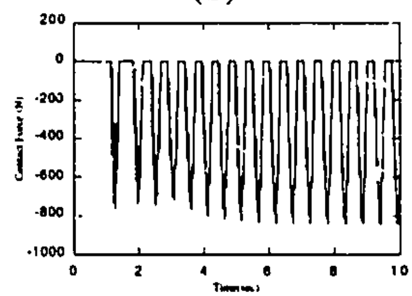
Figure D.5: Performance of the MIC, k_p =1000, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.



(a)

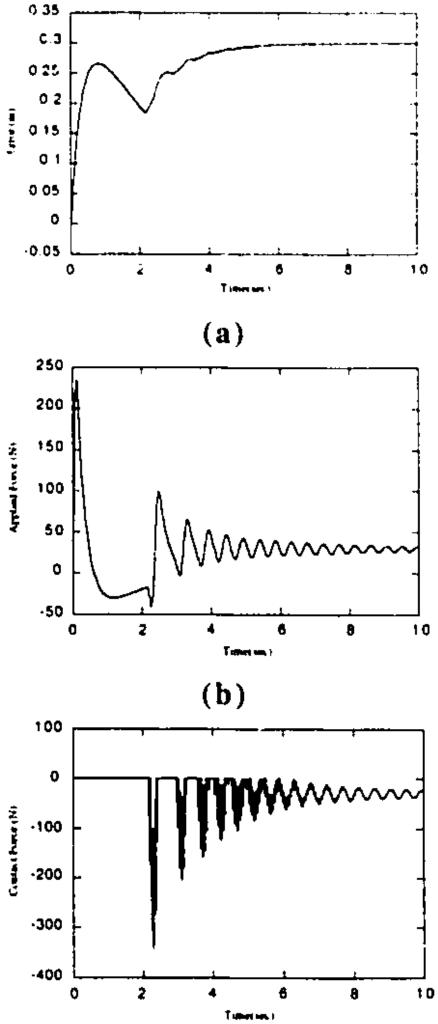


(b)



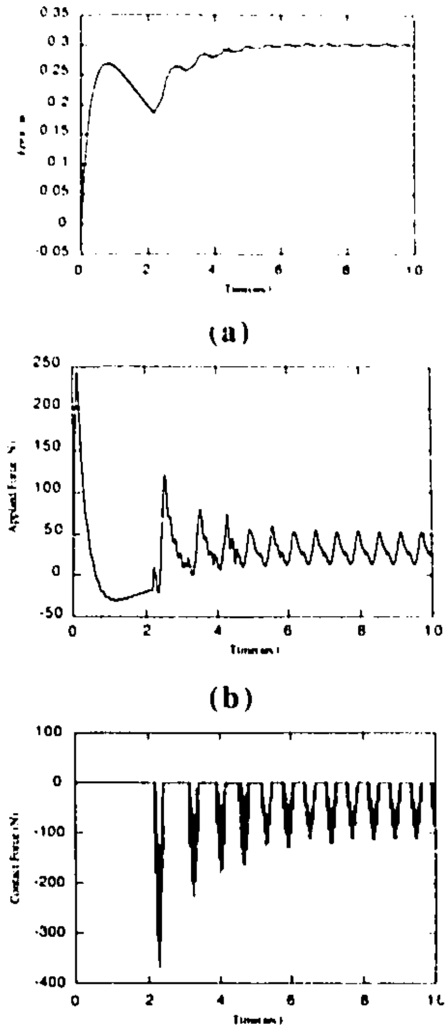
(c)

Figure D.6: Performance of the OIC, k_p =1000, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.



(a) (b) (c)

Figure D.7: Performance of the MIC, b_2 =10, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.



(a) (b) (c)

Figure D.8: Performance of the OIC, b_2 =10, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.

Figures D.7 and D.8 compare the simulated performance of the MIC and OIC for a relatively lower damped object, i.e. $b_2 = 10.0 kg/\sec$. The rest of the system parameters are the same as before, and so are the controller parameters, i.e. $m_{des} = 100.0$, $k_p = 100.0$, and $k_d = 300.0$. It can be seen that the performance of both algorithms does not show any considerable deterioration. It is interesting to note that the root locus analysis shows that for the given parameters, the system under the OIC law becomes unstable in the "no contact" phase.

As discussed before, choosing larger values for the desired mass besides larger k_d, can result in a stable system, see Appendix D. In this case (for $b_2 = 10.0 kg/\sec$), m_{des} =500, and k_d =1000 guarantees the stability of the OIC in both phases. Figure D.9 shows the effect of these choices on the performance of the OIC. As it is seen, choosing larger values for the desired mass makes the system sluggish, where large k_d's can reasonably damp the oscillations.

Based on the investigations presented in this appendix, it can be concluded that the MIC algorithm yields a preferable performance, compared to the OIC.

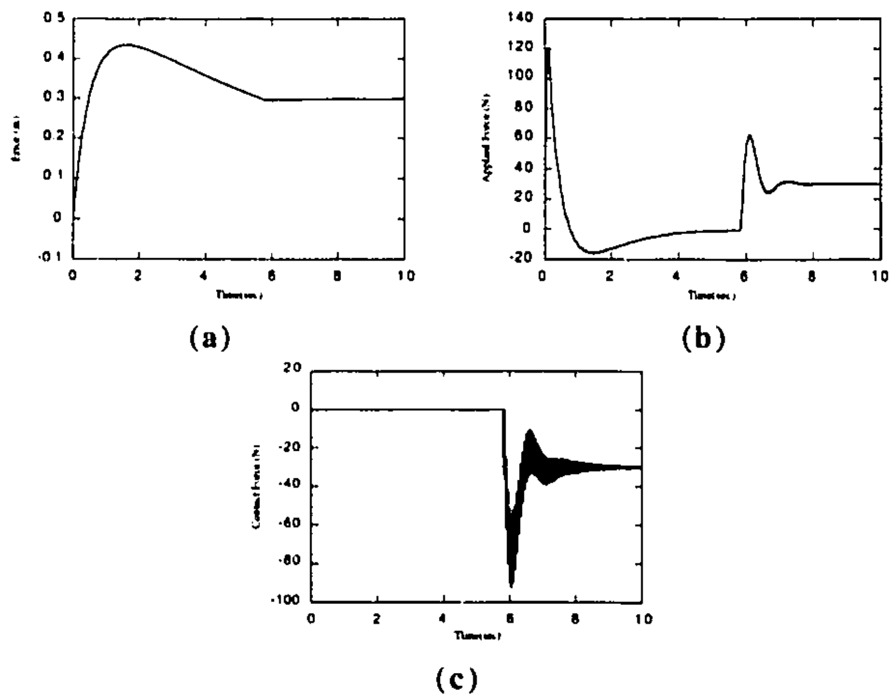


Figure D.9: Performance of the OIC, for b_2 =10, m_{des} =500, k_d=1000, (a) Object tracking error, (b) The applied controlling force, (c) The contact force.