NETWORKED MULTI-MANIPULATOR SYSTEM AND ITS TELEOPERATION USING ADAPTIVE NON-SINGULAR TERMINAL SLIDING MODE CONTROL

by

Henghua Shen

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
October 2020

Table of Contents

List of Tables . . . . **vi**

List of Figures . . . . **vii**

Abstract . . . . **xii**

List of Abbreviations and Symbols Used . . . . **xiii**

Acknowledgements . . . . **xxii**

Chapter 1 Introduction **1**

1.1 Robot Manipulation Systems 1

1.2 Multi-Agent Systems 3

1.3 Teleoperation Systems 4

1.4 Research Motivation 6

1.5 Thesis Outline and Contributions 9

Chapter 2 Literature Review **12**

2.1 Sliding Mode Control 12

2.2 Control of Multi-Manipulator Systems 13

2.3 Control of SMMS Systems 16

Chapter 3 Background Theories **19**

3.1 Robotics Terminology 19

3.2 Kinematic Model 20
  3.2.1 Homogeneous Transformations 20
  3.2.2 Denavit-Hartenberg Convention 21
  3.2.3 Jacobian Matrix 23
  3.2.4 Unit Quaternion 25
  3.2.5 Examples 27

3.3 Dynamic Model 29
  3.3.1 Joint Space Dynamics 30
  3.3.2 Task Space Dynamics 30

3.3.3 Examples 32

3.4 Network Communications 34
    3.4.1 Network Representation 34
    3.4.2 Directed Graph Connectivity 34
    3.4.3 Network-Induced Constraints 35

3.5 Non-Singular Terminal Sliding Mode Control Theory 38

Chapter 4 Synchronization Control Design for Leader-Following Manipulators under Network-Induced Constraints 40

4.1 Problem Description 40

4.2 NTSM Control Method Using Mixed-Type Feedback 42
    4.2.1 Types of Feedback Signals 43
    4.2.2 Controller Development 44
    4.2.3 Simulation Results 48
    4.2.4 Experimental Results 57

4.3 Cooperation Control Design with Online Neighbor Selection 61
    4.3.1 Continuous-Discrete Unscented Kalman Filter 62
    4.3.2 Noise Estimation 65
    4.3.3 Neighbor Selection Policy 65
    4.3.4 NTSM Controller Design 68
    4.3.5 Simulation Results 69

4.4 Concluding Remarks 74

Chapter 5 Adaptive NTSM Control with Time-varying Gains . . 75

5.1 Problem Description 75

5.2 Adaptive NTSM Control with Time Varying Gains 77
    5.2.1 Mixed-Type Feedback Signals 77
    5.2.2 Compensatory Bounds 77
    5.2.3 Adaptive NTSM Control 80
    5.2.4 Time-Varying Gain Design 81

5.3 Discussions on Control Gain κ_i 83
    5.3.1 Results of Constant Gain Control 83
    5.3.2 Results of the Time-varying Gain control 83

5.4 Illustrative Examples 90
    5.4.1 Simulation Results 90
    5.4.2 Experimental Results 93

5.5 Concluding Remarks 97

Chapter 6 Bilateral Teleoperation Control Design 98

6.1 Problem Descriptions 98

6.2 Control Development 99
  6.2.1 Environmental Force Prediction 100
  6.2.2 Master Controller Design 101
  6.2.3 Follower Controller Design 102

6.3 Numerical Simulations 104
  6.3.1 Comparison of Different Types of Feedbacks 105
  6.3.2 Tracking Performance Illustration 106
  6.3.3 Comparisons with existing control methods 108

6.4 Experimental Results 111

6.5 Concluding Remarks 114

Chapter 7 Teleoperation of Multiple Cooperative Slave Manipulators 115

7.1 Problem Description 115

7.2 Controller Development 118
  7.2.1 Master Controller Design 118
  7.2.2 Slave Controller Design 121

7.3 Simulation Results 128
  7.3.1 Performance of the Master Manipulator 129
  7.3.2 Performance of the Cooperative Slave Manipulators 132

7.4 Concluding Remarks 136

Chapter 8 Conclusions and Future Work 137

8.1 Conclusions 137

8.2 Future Work 139

Bibliography 141

Appendix A Theoratical Proof in Chapter 4 151

A.1 Proof of Theorem 4.1 151

A.2 Proof of Proposition 4.1 153

A.3 Stability Proof of Theorem 4.3 154

Appendix B Theoratical Proof in Chapter 5 157

B.1 Proof of Theorem 5.1 157

B.2 Proof of Theorem 5.2 159

Appendix C Theoratical Proof in Chapter 6 161

C.1 Proof of Theorem 6.1 161

C.2 Proof of Theorem 6.2 162

C.3 Proof of Theorem 6.3 164

C.4 Proof of Proposition 6.1 166

C.5 Proof of Proposition 6.2 166

Appendix D Transition Waveform 167

Appendix E Author's Publications 169

List of Tables

3.1 D-H parameters for 2-DOF planar manipulator. 27

3.2 D-H parameters of the Phantom Omni device 29

3.3 Dynamic parameters of the Phantom Omni device 34

3.4 Mechanical limits of the Phantom Omni device 34

4.1 The maximum assigned velocity and acceleration generated by different angular frequencies f. 49

4.2 Results of tracking errors affected by network delays 52

4.3 Results of tracking errors affected by angular frequencies 52

4.4 Results of tracking errors affected by α. 54

4.5 Results of tracking errors affected by β. 54

4.6 The convergence time 55

5.1 The oscillation and settling time with different constant κ_i. . . 83

5.2 The oscillation subject to the intial switching gain $\kappa_i(0)$ 88

5.3 The oscillation subject to the maximum value L_1 88

5.4 The oscillation subject to the maximum value L_2 89

6.1 Comparisons of three control methods 109

6.2 Steady-state errors using three different controllers 109

List of Figures

1.1 Examples of robot manipulators with different DOFs 1

1.2 Examples of industrial application using robot manipulators . 2

3.1 Schematic of a two-link planar robot manipulator illustrating the essential robotics terminology. 20

3.2 Coordinate frame assignment and D-H parameters of two general manipulator links 22

3.3 Coordinate frame assignment of a two-link planar manipulator. 27

3.4 Schematics and picture of the Phantom Omni haptic device. . 28

3.5 Examples of different types of directed graph connectivity . . 35

3.6 Schematic of a network control system of two agents with time-varying delays and the packet loss. 36

3.7 Network delay measurements of the wired LAN and wireless LAN 38

4.1 Weakly connected topology. 49

4.2 Comparison of the translational tracking performance using different types of feedback signals 50

4.3 The Box plot shows the medians, quartiles, extremes, outliers, and the average of the pose tracking error varying with delays and angular frequencies 52

4.4 Selection guideline of the gain α 53

4.5 The Box plot shows the medians, quartiles, extremes, outliers, and the average of the pose tracking error varying with α and β values 54

4.6 Plots of the convergence to the auxiliary switching surface . . 56

4.7 The convergence of the sliding surface. 56

4.8 The pose tracking synchronization of the illustrative example . 57

4.9 The norm of the translational and angular position tracking errors 57

4.10 The experimental testbed. 58

4.11 Measured one-way network delays in a wired LAN by manually adding a delay of 0.03 s 58

4.12 Control inputs of the two followers. 59

4.13 The translational position tracking and tracking errors with time-varying reference signals. 59

4.14 The angular position tracking and tracking errors with time-varying reference signals. 60

4.15 The unfiltered and filtered velocities and accelerations 60

4.16 Leader-following network topology. 69

4.17 The CD-UKF has a good ability to obtain the smooth average state from the noisy measuring signals. 70

4.18 (a) The convergence of estimated measurement noise levels to their actual noise levels. (b) Comparisons of estimated process noise levels to their actual noise levels 71

4.19 (a) The mechanical energy evolution and (b) tracking error (in x direction) of the 3rd manipulator in **Case 1** and **Case 2**. . . 72

4.20 The mechanical energy evolution and tracking errors of the 3rd manipulator in Case 3 and Case 4 72

4.21 Energy indexes of the manipulator $1 \sim 4$ and selection indexes of the agent 3 73

5.1 (a) Position profiles with different fixed κ_i; (b) The relation between κ_i and steady-state error Ex_{ss} 84

5.2 (a) Comparison of the tracking results using time-varying gain and constant gains; (b) Two aberration regions 84

5.3 The effects of the Sigmoid midpoint, logarithm base, and growth rate on the establishment time 85

5.4 (a) The effect of the establishment time on the duration of the pre-transition aberration region; (b) Relation of establishment time and the settling time under different initial errors. 86

5.5 (a) Tracking results with different maximum values L_2; (b) Relationship between L_2 and the steady-state error. 87

5.6 Weakly connected topology of the leader-following manipulator system. 91

5.7 Time-varying network delays and external torque disturbances 91

5.8 Position tracking of the end effectors 92

5.9 Position tracking errors between the followers and the leader by using (a) sign function, and (b) saturation function in the follower controllers. 92

5.10 The resultant friction of follower manipulators using (a) sign function, and (b) saturation function in the controller. 92

5.11 (a) The adaption of the unknown parameter $\hat{\Theta}_i$; The compensatory bounds for (b) for velocity error caused by self-delay; (c) for acceleration error caused by time-varying delays; (d) for acceleration estimating error. 93

5.12 Testbed of the networked leader-following system with a pair of Phantom Omni haptic devices. 94

5.13 Position tracking and the tracking errors of the leader-following system with time-varying human-commanded reference signals. 95

5.14 Filtered control inputs, the compensatory bounds, and estimation of the unknown parameters 95

5.15 Tracking performance comparison amongst Fuzzy-SMC method, RBFNN-ASMC method, and the proposed ANTSM approach 96

6.1 Bilateral teleoperation control architecture. 100

6.2 The end effector position tracking synchronization of the master and the slave with controllers using different types of feedback 105

6.3 Plots of network delays, external disturbances, and frictional forces 106

6.4 Plots of the compensatory bounds and estimation of the unknown parameters 107

6.5 The position tracking and tracking errors with constant reference signals as in Case 1 108

6.6 The position tracking and tracking errors with time-varying reference signals as in Case 2 108

6.7 Environmental force estimations for Case 1 and Case 2 109

6.8 The master-slave position tracking in Cartesian space with time-varying reference signals using P-like controller 110

6.9 The master-slave position tracking in Cartesian space with time-varying reference signals using a model-based NTSM controller. 110

6.10 The teleoperation experimental testbed with two network-connected Phantom Omni haptic devices. 112

6.11 Position tracking and the tracking error of the master-slave system with time-varying reference signal. 113

6.12 Control inputs for joint motors of the master and slave manipulators 113

6.13 The compensatory bounds for the slave controller 113

6.14 Plots of (a) the estimated environmental forces and the true environmental forces, and (b) the estimation of the unknown dynamic parameters 114

7.1 Control architecture of an SMMS manipulation system. 116

7.2 Grasp kinematics of the cooperative manipulation system. . . 122

7.3 Schematic of the SMMS manipulation system using the Phantom Omni haptic devices. 129

7.4 Environmental wrench prediction and comparison to the true values 130

7.5 Translational position tracking of the master end effector towards the generated reference signals. 131

7.6 Angular position tracking of the master end effector towards the generated reference signals. 131

7.7 Position tracking errors and the resultant internal forces. . . . 132

7.8 Orientation controller errors and the resultant internal torques. 132

7.9 Translational position tracking of the leader end effector towards the master manipulator's translational trajectory. 133

7.10 Angular position tracking of the leader end effector towards the master manipulator's orientation. 133

7.11 Wrench distribution with the weighting factors given in Case 1 134

7.12 Wrench distribution with the weighting factors given in Case 2 135

7.13 The produced additional torques in (a) Case 1 and (b) Case 2 are respectively compensated by the allocated torques shown in (c) and (d) from the wrench allocation policy 135

D.1 The positive-going transition waveform. 167

Abstract

Teleoperation of multiple robot manipulators has been one of the most popular research areas in the robotics research community for the last couple of decades. Such complex systems can be decoupled into two subsystems, namely, multi-agent systems (MASs) and teleoperation systems.

In addition to the high nonlinearity of the networked multi-manipulator systems, deleterious effects, caused by network-induced constraints and the lack of exact robot modelling information, can make the control systems' desired performance and stability difficult to achieve. To meet these challenges, concepts from the non-singular terminal sliding mode (NTSM) control method are developed to achieve the exogenous disturbance rejection and the finite-time full-pose synchronization. Additionally, a new adaptive NTSM (ANTSM) scheme is designed for multi-manipulator systems where the models may be initially uncertain or slowly varying over time. To further improve the performance, a set of novel techniques are developed, including the use of novel mixed-type feedback, time-varying logistic-function-based control gain, and energy-index-based neighbour selection policy. The proposed ANTSM approach has also successfully been applied to the teleoperation control systems. In addition, the master manipulator uses a force predictor to estimate the real-time environmental force on the slave side so that the direct transmission of the force signals is avoided.

The proposed approaches for the MASs and teleoperation subsystems are integrated into a single-master-multiple-slave manipulator system. Simulation and experimental results validate the efficacy of the proposed schemes.

List of Abbreviations and Symbols Used

Acronym

ANTSM	Adaptive Non-singular Terminal Sliding Mode.
AST	Active Switching Topology.
CD-UKF	Continuous-Discrete Unscented Kalman Filter.
CoM	Center of Mass.
D-H	Denavit-Hartenberg.
DOF	Degree(s) of Freedom.
E-L	Euler-Lagrange.
Fuzzy-SMC	Fuzzy Sliding Mode Control.
IP	Internet Protocol.
IQR	Interquartile Range.
IWD	Inverse Wishart Distribution.
LAN	Local Area Network.
MASs	Multi-Agent Systems.
MIMO	Multi-Input-Multi-Output.
MMMS	Multiple-Master-Multiple-Slave.
MMSS	Multiple-Master-Single-Slave.
NTSM	Non-singular Terminal Sliding Mode.
PCWF	Point-Contact-With-Friction.
P-like	Proportional-like.
PD	Proportional-Derivative.
PRL	Percent Reference Level.
PST	Passive Switching Topology.
R_{post}	Post-Transition Aberration Region of a Wave Form.
R_{pre}	Pre-Transition Aberration Region of a Wave Form.
RBFNN-ASMC	Radial Basis Function Neural Network based Adaptive Sliding Mode Control.

RFI	Radio Frequency Interference.
SMC	Sliding Mode Control.
SMMS	Single-Master-Multiple-Slave.
UDP	User Datagram Protocol.
UKF	Unscented Kalman Filter.

Chapter 3

A	Adjacency matrix indicating the network connectivity between the followers.
b	Adjacency matrix indicating the network connectivity between the leader and the followers.
$\boldsymbol{q}$	Angular position vector of a robot manipulator's joints.
$\boldsymbol{w}$	Angular velocity.
$\boldsymbol{\xi}^*$	Conjugate of $\boldsymbol{\xi}$.
$\mathbb{C}$	Coriolis and centrifugal loading matrix in joint space.
C	Coriolis and centrifugal loading matrix in task space.
$\bar{C}$	Known nominal Coriolis and centrifugal loading matrix in task space.
$\boldsymbol{\tau}$	Control input in joint space.
$\boldsymbol{u}$	Control input in task space.
$\boldsymbol{\tau}^e$	Equivalent environment torque exerted on the manipulator.
$T_{ij}(t)$	Equivalent delays resulting from network delays and packet loss.
$\boldsymbol{f}^f$	Frictional force.
$\mathbb{G}$	Gravitational loading vector in joint space.
$\boldsymbol{g}$	Gravitational loading vector in task space.
T_1^2	Homogeneous transformation from the frame $o_1x_1y_1z_1$ to the frame $o_2x_2y_2z_2$.
J_p, J_o	Jacobian matrices for translational and angular operations, respectively.

$\bar{\boldsymbol{g}}$	Known nominal gravitational loading vector in task space.
Φ	Known regressor of the uncertain model parameter $\boldsymbol{\rho}$.
$\bar{M}$	Known nominal symmetric and uniformly positive definite inertia matrix in task space.
$\bar{T}^{\mathrm{v}}_{ij}$	Known and constant nominal network delays from agent j to i.
E	Mapping matrix between a geometric Jacobian matrix and an analytical Jacobian matrix.
G	Network graph.
$\mathbb{S}_{ij}(t_k)$	Network states from agent j to i at time t_k.
α, β, κ	NTSM controller gains.
$\boldsymbol{\xi}$	Orientation represented by unit quaternion.
$\boldsymbol{e}_{\xi}$	Quaternion error.
R_1^2	Rotation matrix from the frame $o_1x_1y_1z_1$ to the frame $o_2x_2y_2z_2$.
$\mathrm{sat}(\cdot)$	Saturation function.
$\mathrm{sgn}(\cdot)$	Signum function.
$S(\bullet)$	Skew-symmetric matrix operation.
$\mathbb{M}$	Symmetric and uniformly positive definite inertia matrix in joint space.
M	Symmetric and uniformly positive definite inertia matrix in task space.
$\boldsymbol{d}$	Translational distance from origin o_1 to origin o_2.
$\boldsymbol{p}$	Translational position in a three-dimensional coordinate system.
$T^{\mathrm{v}}_{ij}(t)$	Time-varying network delays from agent j to i.
ΔC	Uncertain portion of the Coriolis and centrifugal loading matrix in task space.
δ^{v}_{ij}	Unknown and bounded delay deviation from the nominal delays.

$\Delta \boldsymbol{g}$	Uncertain portion of the gravitational loading vector in task space.
ΔM	Uncertain portion of the symmetric and uniformly positive definite inertia matrix in task space.
$\boldsymbol{\rho}$	Uncertain model parameter of a robot manipulator.
Θ	Unknown portion of the uncertain model parameter $\boldsymbol{\rho}$.

Chapter 4

$\boldsymbol{e}^{ap}$	Angular position feedback.
$\boldsymbol{e}^{av}$	Angular velocity feedback.
K	CD-UKF gain.
T_{ij}	Constant network delays from agent j to agent i.
$\boldsymbol{\omega}$	Covariance matrix estimated from the inverse Wishart distribution.
$\bar{\boldsymbol{\xi}}$	Estimated orientation of $\boldsymbol{\xi}$.
$\boldsymbol{q}$	Generalized coordinates of a robot manipulator.
$G(s)$	Low-pass filter.
$H_m(\cdot)$	Measurement model.
D	Measurement model coefficient.
R_k	Measurement noise covariance matrix.
$\boldsymbol{Y}$	Measurement of a manipulator's states.
$\boldsymbol{r}$	Measurement noise vector.
Ω	Neighbour selection index.
$\boldsymbol{\varpi}$	Normalized stored mechanical energy.
$\boldsymbol{\xi}$	Orientation (represented by unit quaternion) of a manipulator's end effector.
$\boldsymbol{x}$	Pose of a manipulator's end effector.
$\boldsymbol{e}$	Position feedback.
Δ_x	Position tracking error between the network-delayed leader and a follower.

$\bar{P}y^-$	Predicted measurement covariance matrix.
Px^-	Predicted covariance matrix of $\boldsymbol{X}$.
$\bar{\boldsymbol{X}}^-$	Predicted mean value of $\boldsymbol{X}$.
$\hat{\boldsymbol{Y}}^-$	Predicted mean measurement.
χ^-	Predicted sigma point.
$V(t_k)$	Process noise covariance matrix.
$F_m(\cdot)$	Process model of a manipulator.
$\boldsymbol{X}$	Process model state configured for the CD-UKF algorithm.
M^+	Pseudo-inverse matrix of the matrix M.
Φ	Rate of the inverse Wishart distribution.
$\boldsymbol{T_{ns}}$	Sampling time of sensor measurements.
$sat(\cdot)$	Saturation function.
$\dot{\boldsymbol{e}}^{lc}$	Self-delayed translational velocity error.
ϵ	Shape of the inverse Wishart distribution.
χ	Sigma point.
$\boldsymbol{s}$	Sliding surface.
$\boldsymbol{E}$	Stored mechanical energy.
$\boldsymbol{p}$	Translational position of a manipulator's end effector.
$\boldsymbol{e}^{lp}$	Translational position feedback.
$\boldsymbol{e}^{lv}$	Translational velocity feedback.
$T_{ij}(t)$	Time-varying network delays from agent j to agent i.
A^{aM}	Upper bound of angular acceleration.
V^{aM}	Upper bound of angular velocity.
A^{lM}	Upper bound of translational acceleration.
V^{lM}	Upper bound of translational velocity.
$\mathbb{L}$	Upper bound of velocity estimation error.
Px^+	Updated covariance matrix of $\boldsymbol{X}$.
$\bar{\boldsymbol{X}}^+$	Updated mean value of $\boldsymbol{X}$.
Δ_ϵ	Upper bound of position tracking error.

W^m, W^c, W^{mc}	Unscented transform weights.
$\dot{\boldsymbol{e}}$	Velocity feedback.

Chapter 5

$\boldsymbol{\varepsilon}^{tv}$	Acceleration Estimating error.
$\boldsymbol{\varepsilon}^{ae}$	Acceleration Estimating error.
$\bar{T}$	Average delay.
$\mathbb{B}^{sd}$	Compensatory bound for the self-delay-induced velocity error $\boldsymbol{\varepsilon}^{sd}$.
$\mathbb{B}^{ae}$	Compensatory bound for the acceleration estimating error $\boldsymbol{\varepsilon}^{ae}$.
$\mathbb{B}^{tv}$	Compensatory bound for the delay-variation-induced acceleration error $\boldsymbol{\varepsilon}^{tv}$.
L_1	Curve's maximum value associated with a waveform's pre-transition region.
L_2	Curve's maximum value associated with a waveform's post-transition region.
L	Curve's maximum value of a logistic function.
$\hat{\ddot{\boldsymbol{x}}}$	Estimated acceleration.
$\hat{\Theta}$	Estimation of the unknown modelling parameter θ.
T_{etm}	Establishment time.
Φ	Known state-dependent regressor.
$f(t)$	Logistic function.
p	Logistic growth rate of a logistic function.
a	Lower asymptote of a logistic function.
e	Natural logarithm base of a logistic function.
$\boldsymbol{e}^p$	Position feedback.
$\boldsymbol{\varepsilon}_i^{sd}$	Self-delayed velocity error.
t_{sm}	Sigmoid's midpoint of a logistic function.
Ex_{ss}	Tracking error during steady-state phase.

$T_{ij}(t)$	Time-varying delay from agent j to agent i.
$\kappa(t)$	Time-varying ANTSM gain.
Θ	Unknown modelling parameter.
$\delta_{ij}(t)$	Unknown delay deviation from the average delay $\bar{T}$.
$\boldsymbol{e}^v$	Velocity feedback.

Chapter 6

$\boldsymbol{\varepsilon}^{sd}$	Auxiliary self-delayed velocity error.
$\mathbb{B}^a$	Compensatory bound for the delay-variation-induced acceleration error.
$\mathbb{B}^{sd}$	Compensatory bound for the auxiliary self-delayed velocity error.
$\boldsymbol{x}$	End effector position.
$\boldsymbol{f}_e$	Environmental force.
$\hat{\Theta}_e$	Estimation of the unknown environmental force parameter.
$\hat{\boldsymbol{f}}_e$	Estimated environmental force.
$\hat{\Theta}_m, \hat{\Theta}_s$	Estimation of the unknown modelling parameter θ_m for the master controller and θ_s for the slave controller, respectively.
t_s	Finite time to reach the sliding surface.
$\boldsymbol{f}_h$	Force given by the human operator.
$\boldsymbol{e}_m^p$	Position error between the master state and the reference signal.
$\boldsymbol{e}_s^p$	Position error between the master and slave manipulators.
Φ_e	Regressor of the environmental force.
$\boldsymbol{s}$	Sliding surface.
$M_r, C_r, \boldsymbol{g}_r$	Target parameters of the master behaviour.
Θ_e	Unknown portion of the environmental force.
Θ_m, Θ_s	Unknown modelling parameter for the master and slave controllers, respectively.
ϵ	Upper bound of tracking error between the slave and master manipulators.

$\boldsymbol{e}_m^v$	Velocity error between the master state and the reference signal.
$\boldsymbol{e}_s^v$	Velocity error between the master and slave manipulators.

Chapter 7

$\dot{\boldsymbol{x}}_d$	Stacked vector of the allocated desired slave velocity in task space.
$\boldsymbol{\omega}$	Angular velocity.
$\boldsymbol{\xi}$	Angular position (orientation).
$\dot{\boldsymbol{\varepsilon}}^{sd}$	Auxiliary self-delayed velocity error.
$\mathbf{O}$	Center of mass (CoM) of the object.
C_λ	Compensation matrix for wrench distribution.
$\boldsymbol{q}_d$	Desired angular signal in joint space.
Δ	Displacement from the physical CoM to the virtual CoM.
ζ_i	Distribution coefficient.
$\boldsymbol{x}$	End effector pose.
$\hat{\Theta}_{0,e}$	Estimation of the unknown environmental wrench parameter $\theta_{0,e}$.
$\hat{\Theta}_0$	Estimation of the unknown leader modelling parameter θ_m.
$\hat{\Theta}_m$	Estimation of the unknown master modelling parameter θ_m.
$\boldsymbol{h}_{int}$	Estimation of internal wrench.
$\boldsymbol{f}_h$	Force exerted by the human operator.
$\boldsymbol{f}_i$	Force exerted by the i^{th} slave manipulator.
G^+	Generalized inverse of the grasp matrix G.
G	Grasp matrix.
g	Gravitational acceleration.
J	Jacobian matrix.
$\Phi_{0,e}$	Known slave-state-dependent regressor for the environmental wrench estimation.

Φ_m	Known master-state-dependent regressor for the environmental wrench estimation.
$\boldsymbol{\xi}_{\mathbf{O}}^{i}$	Orientation of frame $\{O_{\mathbf{O}}X_{\mathbf{O}}Y_{\mathbf{O}}Z_{\mathbf{O}}\}$ with respect to the coordinate $\{O_iX_iY_iZ_i\}$ of the i^{th} grasp point.
$\boldsymbol{\xi}_{\mathbf{O}}$	Orientation of $\mathbf{O}$ with respect to the world frame $\{O_wX_wY_wZ_w\}$.
$\boldsymbol{\xi}_i$	Orientation of the grasp point of i^{th} manipulator with respect to the world frame $\{O_wX_wY_wZ_w\}$.
M_r, C_r	Parameters of the target impedance model.
$\boldsymbol{e}_m$	Pose tracking error.
$\boldsymbol{x}_{\mathbf{O}}$	Pose of the object's CoM.
$\boldsymbol{p}_{\mathbf{O}}$	Position of $\mathbf{O}$ with respect to the world frame.
$\boldsymbol{r}_i$	Position of the CoM with respect to the coordinate $\{O_iX_iY_iZ_i\}$ of the i^{th} grasp point.
$\boldsymbol{p}_i$	Position of the grasp point of i^{th} manipulator with respect to the world frame $\{O_wX_wY_wZ_w\}$.
$\boldsymbol{e}_0$	Position tracking error for the leader controller design.
$\boldsymbol{q}_r$	Reference angular signals in joint space.
$\boldsymbol{x}_r$	Reference signal in task space.
$R_i^{\mathbf{O}}$	Rotation matrix of frame $\{O_iX_iY_iZ_i\}$ with respect to the frame $\{O_{\mathbf{O}}X_{\mathbf{O}}Y_{\mathbf{O}}Z_{\mathbf{O}}\}$.
$\boldsymbol{h}_i$	Wrench exerted by the i^{th} slave manipulator.
$\boldsymbol{h}_{\mathbf{O}}$	Task-space wrench exerted on the slave end effectors by the the object.
λ_i	Torque compensation coefficient.
$\boldsymbol{t}_h$	Torque exerted by the human operator.
$\boldsymbol{t}_i$	Torque exerted by the i^{th} slave manipulator.
$\boldsymbol{p}$	Translational position.
$\boldsymbol{h}_h$	Wrench exerted by the human operator.

Acknowledgements

This thesis was completed with financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the China Scholarship Council (CSC) and the Nova Scotian Government.

I would like to thank my supervisor, Dr. Ya-Jun Pan, for all of her support. She continuously provided encouragement and was always enthusiastic to assist in any way she could through my time at Dalhousie. I would also like to thank my supervisory committee members, Dr. Mae Seto, Dr. Serguei Iakovlev, and Dr. Wen-Fang Xie, for their feedback and advice.

I am grateful to my fellow researchers at the Advanced Control and Mechatronics Laboratory for their willingness to help towards my thesis. I would also like to extend gratitude to friends outside the lab for their emotional and spiritual support.

Finally, I would like to thank my parents and sisters. They have been with me every step of the way and I could not have done it without their love and support.

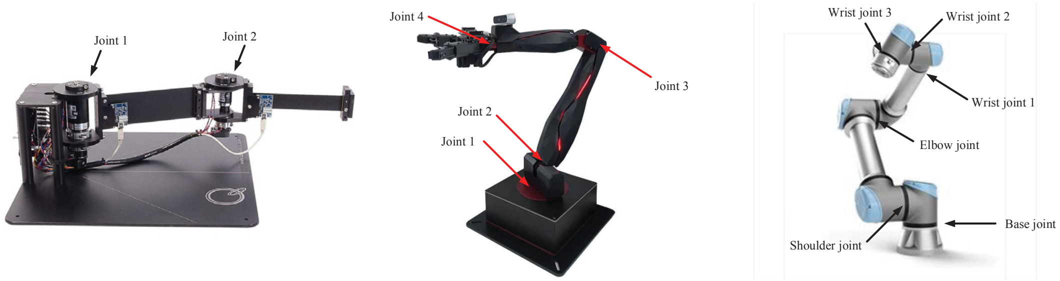
Chapter 1

Introduction

In this chapter, we will introduce some basic information and necessity to study the cooperation and teleoperation of robot manipulator systems.

1.1 Robot Manipulation Systems

A robot manipulator (or robotic arm) is an electronically controlled mechanism used to manipulate materials without direct physical contact by the human operator. The arm-like mechanism usually consists of a series of sliding or jointed segments with one fixed end and one free end to perform a given task. In this text, the robot manipulators will refer to those of rigid bodies. There are different criteria to categorize robot manipulators. One common way to classify robot arms is by the number of degrees of freedom (DOFs) determined by the number of joints [1]. Fig.1.1 shows three examples of serial manipulators classified by DOFs with their joints indicated. A manipulator should possess at least six independent DOFs in order to reach an arbitrary pose (translation position and orientation) in its work environment.

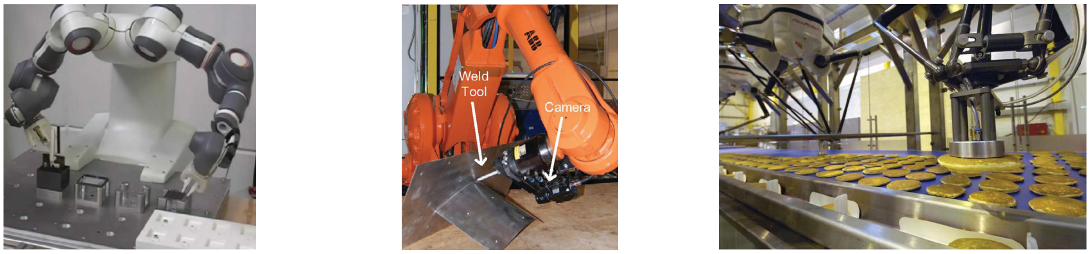


(a) 2-DOF (b) 4-DOF (c) 6-DOF

Figure 1.1: Examples of robot manipulators with different DOFs: (a) QUANSER 2 DOFs Serial Flexible Link [2] ; (b) QUANSER QArm 4 DOF serial robotic manipulator [3]; (c) Universal UR3e 6 DOFs robotic manipulator [4].

Robot manipulators are extensively used in the industrial setting for tasks that are repetitive or exceed the speeds and accuracies of human operators. Applications vary from assembly tasks in the auto industry to picking and packaging tasks in the food industry. In those applications, manipulators are not required with high autonomous ability, and, therefore, are reliable and easy-to-use. Three examples of the industrial manipulator applications are shown in Fig.1.2. Robot manipulators also have many other specialized applications that require a higher level of autonomy. In recent years, techniques of multi-sensing systems and artificial intelligence (AI) developments, like machine learning (ML) and deep learning, have significantly improved the autonomy of robotic arms in performing complex tasks. For example, household robots assist people who need daily physical assistance in getting dressed [5].

Industrial robots (including other types of robots) play a critical role in economic development worldwide. The preliminary statistics of new World Robotics Report shows that a new record high of (first time) over 400,000 units were shipped globally as of 2019, and this number is expected to increase to 584,000 units by the year of 2022, an increase of +12% on average per year [6]. In 2020, the Coronavirus (COVID-19) pandemic was reportedly predicted to cause a global economic crisis, and industrial robots are believed to play a vital role in automating production to speed up the post-Corona economy. Therefore, research on industrial robotics continues to be of significant economic meaning in the near future.



(a) Assembly parts (b) Welding automation (c) Cheese picking

Figure 1.2: Industrial applications of robot manipulators: (a) ABB FRIDA dual-armed robot for assembly tasks [7]; (b) An ABB IRB2400-robot for welding tasks [8]; (c) The PixCell robotic picking unit for the cheese packing [9].

The study of robot manipulators involves dealing with the positions and orientations of the several segments that make up the manipulators. Generally, research on robot manipulator includes topics such as hardware configurations, mathematical modelling, control systems, sensing systems, and path planning strategies. Also, new research interests keep emerging as robot manipulators have been innovatively used in many interdisciplinary research areas to improve productivity and safety.

1.2 Multi-Agent Systems

Multi-agent systems (MASs) is a computerized system composed of multiple agents interacting in the environment. In robotic systems, a group of interconnected robots exchange information in order to collaboratively perform specific tasks that are beyond a single robot's ability. To define the properties of systems of interacting robots, the use of graph theory [10] is seen to be advantageous because of its well-studied definitions and algorithms. During the task implementation, data exchanging on the fly helps share loads and coordinate the motion of robots ensemble. Network communications can be classified into explicit communication, implicit communication and the hybrid of these two communications [11].

- **Explicit communication**: The agents' information is directly given and exchanged through the physical connection that is established using either cable media or wireless media.
- **Implicit communication**: A robot acquires other robots' data by observing with onboard sensors, such as visual devices.
- **Hybrid communication**: Using the combination of explicit communication and implicit communication to gain rich information and reduce the transmission complexity.

However, in real-world networks the robot interactions inherently time-delayed. Network delays existing in the communication channels make one robot react sluggishly to the change of its neighbours' states, and, therefore, the performance in terms of accuracy may degrade. In addition, other network constraints such as weak connectivity, packet loss, and out-of-order packet arrival can lead to undesired performance and even complete failure. Therefore, the robots' stability and performance under network constraints have become essential subjects in studying MASs.

MASs implemented by robot manipulators are usually studied with the goal of consensus, tracking synchronization, or cooperative tasking.

- **Consensus control**: For multiple robot manipulators with arbitrary initializations, consensus control aims at driving the robots' states to reach a common agreement by exchanging information with a certain communication protocol. In the industrial setting, consensus control approaches are usually implemented by mobile manipulators with tasks like transportation, navigation, or obstacle avoidance.

- **Tracking synchronization**: Unlike the consensus control, the tracking synchronization is discussed with a prescribed reference. The objective is then to drive the robots' states towards the given reference signals. One of the tracking synchronization frameworks is in a leader-following structure, which has been broadly applied in applications with specific and preplanned tasks.

- **Cooperative Tasking**: In the cooperative manipulation, robots ensemble may be assigned with different yet cooperative subtasks. By respectively realizing the subtasks through multiple robot manipulators, more complex and delicate tasks can be carried out. However, it introduces more complexity in the control design because they involve more challenging procedures such as motion distribution, load distribution, and internal force minimization.

1.3 Teleoperation Systems

Teleoperation systems involve the interactive manipulations of a master-slave system where the master information is transmitted to instruct the slave manipulation systems in a remote site. The environmental forces (and/or other primary sensory elements such as vision and sound) are fed-back to the master side provide the human operator with the situational awareness of the remote environment. Teleoperation is most commonly associated with robotics, and, together with the telepresence, make up the two major subfields of telerobotics. There is a notable interest in so-called *bilateral teleoperation* where the teleoperation and telepresence are usually studied at the same time. Through the bilateral teleoperation, human operators can control a robot at a distance and, meanwhile, feel present in the remote environment. However, the human operators' experience is inevitably affected by network constraints such as network delays. Therefore, a great deal of effort seeks to improve the operation accuracy and enhance the human operator's instant immersion under time delays.

Bilateral teleoperation has broad industrial applications, such as explosive removal [12], underwater exploration and inspection [13], and medical surgery [14]. Three popular industrial applications are described as follows.

- **Hazardous applications**: Robot manipulators operated by a person from a distance have been used in hazardous environments to minimize the human operator's risk. Hazards may come from the handled objects or the environments. Through the remotely controlled robots, humans are able to work effectively from a safe environment. A large variety of applications have demonstrated the effectiveness of teleoperation, such as the hazardous material handling and transportation (e.g., NBC (nuclear, chemical, biological) operations), landmine detection and eradication, land excavation, and hazardous material [15].

- **Space applications**: Remotely manned robots in space can serve as the assistants of astronauts to perform in planetary and lunar scientific missions. The use of telerobots for space operation significantly reduces the cost and risk in complex environments arising from the weightlessness, vacuum, and extreme temperature. Because of the difficulty of designing fully autonomous space robots, it would be preferable to be supervised by human operators in the space stations or on the ground. However, the inherent communication constraints (such as large communication delays) pose significant challenges for outer space teleoperation, which motivate plenty of developments in terms of the flexibility and efficiency of the robotic process under the communication complexity and extreme environmental conditions.

  Some highly capable telerobotic systems that are already on the market are (1) Dextre, which is a two-armed telemanipulator, is part of the Mobile Servicing System on the International Space Station (ISS) and designed to perform orbital replacement units [16]; (2) Canadarm 2, which is a 17-metre-long robotic arm that is designed to perform Station maintenance and payloads in space [17]; (3) The European Robotic Arm (ERA) is a telerobotic arm with specific tasks like inspecting the station and handling external payloads [18].

- **Medical applications**: Medical telerobotic systems allow procedures, such as surgeries, treatments, and diagnoses, to be conducted by the physicians from a distance. A robot surgical system generally includes at least one surgeon-controlled robot manipulator, a master controller (also called the console), and a sensory system that provides the surgeon with the feeling of being in that other location. On the remote side (or patient side), it includes a robotic system to embody the human gestures to perform medical tasks. Additionally, a multi-sensory integration system collects information of the environment and the tasks and feeds back it to the master side, which provides the telepresence of human operators. Medical telerobotic systems can relieve the burden of surgeon shortage and allows the expertise of specialized surgeons to be available to patients worldwide with less concern about the physical location limitation. The development of sensory systems also improves the safety and accuracy when the area of operation is limited. Some telerobotic medical systems with successful clinical applications are Telelap ALF-X Endoscopic Robotic Surgical System [19], ZEUS robotic surgical system [20], da Vinci Surgical System [21], etc.

1.4 Research Motivation

Single-master-multiple-slave (SMMS) manipulation systems combine the benefits of teleoperation systems and MASs. However, nonlinear control of an SMMS system remains a challenging problem as it involves the coordination of slave arms as well as the real-time interaction between the master side and the slave side. The SMMS systems have less been discussed due to the control complexity. Nevertheless, it is of significant meaning to investigate such systems to improve the robots' load capacity, rigidity, and dexterity.

Information sharing is a critical process in an SMMS system as the promptness and quality of the shared states determine the stability and performance of the system. In practice, however, the transmited data is usually time-varying and time-delayed, which gives rise to adverse consequences such as horizontal shift effect. When the communication network is weakly connected [22, 23], the horizontal shift effect exacerbates as those who have no direct connection with the leader, for example, must rely on the delayed states of their connected neighbours, and, therefore, do not react to the change of the leader until the leader information has been passed through multiple delays. In nonlinear control of the networked multiple slave manipulators, the complete pose regulation has been another challenge. The position dynamics and orientation dynamics are highly coupled, which leads to the difficulty of simultaneously controlling the position and orientation. In the situations where heterogeneous manipulators are used, the quality (e.g., accuracy, noise level, etc) of neighbour signals may vary. As a result, updating a well-performing agent's states based on the low-quality neighbour information can impair the performance of this agent, and, furthermore, that of the overall MASs.

The limitations described above motivate the developments introduced in Chapter 4. The control strategies should be able to solve the following questions to improve the motion synchronization performance of multi-manipulator systems.

- How to efficiently reduce the delay-induced horizontal shift effect to improve motion tracking accuracy?
- How to design a nonlinear controller to realize full-pose synchronization control of a networked multi-manipulator system?
- How to identify a bad-performing neighbour robots and reduce its effect on the otherwise well-performing agents?

Control developments for the SMMS system in this work are based on the concept of the sliding mode control (SMC) method. Conventional SMC approaches that are applied to networked systems heavily rely on the exact mathematical models of the robots and network systems. However, the exact mathematical models is hard to acquire. Therefore, to apply the SMC method to physical robot manipulation systems, the controller should be further developed to deal with unknown network constraints (e.g., time-varying delays), model uncertainties, and external disturbances.

In many sliding mode control approaches, the control gains are usually selected constant and larger than necessary to guarantee the robustness. However, due to the mechanical limitations in many robotic devices (e.g., the limited motor input range), using high switching gains may result in high control energy. Excessive mechanical energy in the system can lead to the significant chattering, oscillations and large tracking errors. On the other hand, small switching gains may not be able to counteract the static friction and, thus, decrease the tracking accuracy in the steady-state phase. Therefore, there is a notable interest in time-varying switching gains in the SMC approaches in many previous work. However, they are usually given as a function of the difference between the current states and the prespecified trajectories. Consequently, in startup operations, large initial tracking errors can result in high control gains that cause significant overshoots/undershoots or large control input that is beyond the motors' input restriction. Unfortunately, the system performance with a time-varying gain during the transient phase has not been well discussed.

The above limitations on the unknown model uncertainties and time-varying control gains motivate the developments in Chapter 5 to address the following questions:

- How to achieve stability and desired motion tracking accuracy for a networked multi-manipulator system where the upper bounds of the time-varying delay and modelling uncertainties are unknown?

- How to design time-varying control gains to obtain a good trade-off between a smooth convergence and a high tracking accuracy in the physical implementation of the ANTSM controller?

The challenging questions for the MASs also apply to the teleoperation subsystem when the ANTSM control approach is used. In addition, telerobotic systems are required to provide the human operators with prompt situation awareness of the remote environment. In this work, we will only consider the haptic sense so that force information from the slave side is required to send to the master side. In the earlier chapters, only the motion consensus control or tracking synchronization control is considered for a multi-manipulator system. Nevertheless, in an SMMS system, considering the coordination of the slave manipulators allows more dexterous operations and more complex tasks, although cooperative control strategies are more challenging to design. Control strategies for SMMS systems are largely based on the successful developments for the MASs and the teleoperation systems. Therefore, in addition to the motivations in the earlier chapters, the two following questions promote the control developments in Chapter 6 and Chapter 7, respectively.

- How to predict the environmental force for the human operators with less affected by network delays?
- How to well distribute the time-varying motion and allocate load to the slave manipulators with a goal of minimizing the internal wrench exerted on the manipulated tasks?

1.5 Thesis Outline and Contributions

This work is organized as follows. Chapter 1 presents the background information on robot manipulator systems, multi-agent systems, and teleoperation systems, followed by the motivations and contributions of this dissertation. Chapter 2 presents the literature review of the existing control schemes for MASs, teleoperation systems, and SMMS systems. Chapter 3 describes the terminology and theory of robot manipulators, network communications, and control methods used in this work. Chapter 4 presents a developed NTSM scheme to ensure the leader-based pose synchronization of a group of nonlinear manipulators that are subject to network delays and weakly connected topology. An novel ANTSM control method is developed for a multi-manipulator system in Chapter 5 and a bilateral teleoperation system in Chapter 6, respectively. In Chapter 7, an SMMS manipulation system is built based on the techniques developed for the multi-manipulator system and the bilateral teleoperation system in the earlier chapters. Chapter 8 summarizes the conclusions of this work and suggests areas for future research.

Contributions of the developments in this work are described as follows.

1. Reducing the phase-shift effect in leader-based pose synchronization using novel mixed-type feedback signals in the NTSM controllers: In a network-delayed leader-following manipulation system, unlike many previous studies where a static leader is assumed [24], we consider a more realistic condition where the leader is dynamical and globally reachable. This, however, introduces a phase-shift effect under network delays. In Chapter 4, we proposed an NTSM control approach to significantly reduce the phase-shift effect using the novel mixed-type feedback signals. In addition, a sufficient condition that ensures the accuracy improvement by using the mixed-type feedback signals is developed in Chapter 5 .

To control the end effector's poses, concepts of the unit quaternion from [25, 26] are used to represent the orientations, which significantly reduces the design complexity of the full-pose tracking synchronization control. Compared to [25, 26] where the orientation regulation represented by unit quaternion is addressed for a single manipulator, in Chapter 4, we successfully apply the unit quaternion representation to the NTSM full pose control for a multi-manipulator system. Besides, multiple effects on the tracking performance in terms of upper error bounds are quantitatively evaluated.

2. Reducing noise propagation using a mechanical-energy-based neighbour selection policy: In a networked multi-manipulator system, we introduce an active neighbour selection policy by monitoring the temporary mechanical energy of the manipulation system. In contrast to the results in [27] and [28] where the active neighbour selection strategy aims to reduce the communication complexity and computation load for MASs of a large number of agents, the real-time active neighbour selection approach in Chapter 4 is designed to reduce the unnecessary noise propagation through a nonlinear, small-volume, and networked multi-manipulator system. This new approach enables a multi-manipulator system to manage neighbour interactions so as to improve tracking accuracy. Additionally, to lessen the dependability on imperfect or unavailable velocity and acceleration readings, a CD-UKF method is successfully used to estimate the manipulator states.

3. Adaptive NTSM control scheme for nonlinear manipulation systems subject to initially unknown parameters: To address the deleterious effects caused by the initially unknown parameters, such as dynamic uncertainties, frictions, external noises, and time-varying network delays, the proposed NTSM controller is further developed into an ANTSM control method. The ANTSM method does not require a priori information about the bounds on these uncertain and time-varying parameters. Additionally, in the proposed ANTSM controller, three novel compensatory bounds are estimated in real-time to compensate for the state errors induced by the system nonlinearity and time-varying network delays. Compared to the most similar work in [29–31] where a basic ANTSM control method is proposed for only one manipulator, we successfully apply the proposed ANTSM scheme to regulate the tracking synchronization of a multi-manipulator system in Chapter 5, the bilateral master-slave teleoperation in Chapter 6, and the implementation of an SMMS manipulation system in Chapter 7. Detailed Lyapunov-based stability and convergence analysis are provided. Simulation results in comparison with other model-free and model-based control methods present the performance improvement of the proposed control approaches. Experimental results of the Phantom Omni haptic devices are also provided to demonstrate the effectiveness of the developed controllers.

4. Obtaining smooth convergence and high tracking accuracy using the ANTSM controller with a logistic control gain function: In the experimental testing in Chapter 5, a constant ANTSM switching gain cannot simultaneously guarantee a smooth convergence in the transient phase and high tracking accuracy. This inadequately addressed issue can be solved efficiently by the proposed time-varying switching gain that is state-independent and designed in a logistic form. Compared to many previous results where the time-varying switching gains in SMC controllers are given as a function of tracking errors for the purpose of chattering reduction and energy saving [30, 31] , the time-varying ANTSM control gain in this work is developed to help the robots with arbitrary initial positions smoothly converge to desired reference signals without significant overshoots/undershoots, and, meanwhile, achieve high steady-state tracking accuracy.

5. Teleoperation control of a cooperative multi-manipulator system using the ANTSM control approach: In Chapter 7, we successfully integrate the developments from the earlier chapters and apply them to an SMMS manipulation system. The ANTSM control approach guarantees the stability and the manipulation objective. Compared to [32, 33] where the SMMS system is constructed to remote control a group of slave robots with simple task motion synchronization, the slave robots of the SMMS system in this work implement the task cooperatively. Therefore, we additionally consider the load distribution to avoid significant internal force exerted on the manipulated object. Compared to [34] where the exact robot model and perfect network communication are assumed for an SMMS manipulation system, the proposed ANTSM control approach can tolerate plant modelling uncertainties and cope with inexactly known and time-varying network delays. Additionally, the use of an optimal wrench allocation strategy can compensate for the additional internal torque exerted on the object.

Chapter 2

Literature Review

This chapter begins with a brief overview of the typical sliding-mode-based control methods with applications to robot manipulators. Additionally, control strategies for multi-manipulator systems and SMMS systems are reviewed with focus towards the challenges most relevant to this work.

2.1 Sliding Mode Control

In the practical robotic control problem, it is inevitable to deal with the effects caused by the discrepancy between the mathematical models and their actual physical systems, such as the plant parametric uncertainties and external disturbances. This has led to intense interest in developing of robust control methods for both linear and nonlinear systems. One of the robust control techniques is the SMC scheme. The control action switching at a high frequency (termed *sliding mode*) is efficient in control of high-order nonlinear dynamic systems in the presence of unknown but bounded noises.

Since the SMC scheme has also been proved to effectively cope with the parametric uncertainties for the complex multi-input-multi-output (MIMO) nonlinear systems, it has been widely studied in robot manipulators with multiple DOFs. Earlier research in [35, 36] outlined the SMC design procedures for the trajectory regulation of multi-joint robot manipulators. [37] shows that using the Euler-Lagrange (E-L) formulation to model robot manipulators greatly facilitate the practical design of sliding mode control scheme, to deal with external disturbances and unstructured model uncertainties, despite the nonlinear interactions and unknown parametric changes. However, a well-known disadvantage of the SMC approach is the chattering effect due to the sliding mode. The high-frequency switching in the control input may damage the robot and decrease the robots' life expectancy. Therefore, extensive research has been made to reduce the chattering effect in the SMC methods [38–40]. However, the benefits of the chattering reduction usually come at the cost of accuracy loss.

The SMC techniques can drive the robots' states towards the desired setpoints or trajectories in a finite time, and the convergence rate is adjustable by tuning the control parameters. However, conventional SMC methods fail to drive their system states to zero in a finite time because of high-frequency switching behaviour. The first attempt can be seen in [36] where the high-frequency chattering effect is effectively reduced by using the continuous control laws to approximate the discontinuous control law. As a result, a smooth convergence of the control states to zero can be realized. An alternative approach to realize the finite-time state convergence to zero is the technique called terminal sliding mode control (TSMC). The TSMC method has been developed for single-input-single-output systems [41] and MIMO systems [42]. Furthermore, the non-singular terminal sliding mode (NTSM) schemes have been developed to control nonlinear rigid robotic manipulator systems to avoid mathematical singularities [42, 43].

Although the NTSM approaches, as well as the conventional SMC methods, make the robotic systems insensitive to the unknown model conditions, the bounds of the unknown conditions are required as a prior. This assumption can be removed by applying the adaptive control technique to the design of the SMC control methods [44]. Z. Man, et al. [29] proposed an ANTSM control method to estimate the boundary of the uncertainty and the disturbance and, therefore, the prior information of the bounds is no longer required. However, the ANTSM control design remains a challenging issue for networked systems in the presence of time-varying delays, parametric uncertainties and disturbances with unknown bounds.

2.2 Control of Multi-Manipulator Systems

MASs have been broadly applied to different research areas. In this section, we focus on reviewing some control challenges of MASs in robotic applications. Three most relevant control issues, namely, phase shift effect, full-pose control, and neighbour selection, are reviewed as follows.

In practice, transmitting data in a networked multi-manipulator system may suffer from significant time delays, leading to adverse consequences, such as the instability and the large internal force/torque. Although the passivity-based control approaches have been used to make the systematic stability insensitive to network delays, assumptions of a constant/static reference signal are usually made to avoid the consideration of the *horizontal shift* effect[1] between the tracking signals [45–47]. Provided a time-varying leader trajectory in a time-delayed and weakly connected leader-following system [22, 23], the horizontal shift effect becomes evident as the followers that have no direct connection with the leader must rely on the delayed states of their connected neighbours. Consequently, those followers cannot react to the leader's change until its information has been passed through multiple delays. Therefore, seeking a new control scheme to alleviate the horizontal shift effect has become necessary in order to improve the robots tracking performance.

Since the assumption of a constant/static leader in network-delayed MASs cannot be met in all applications, some leader-based tracking control approaches with a time-varying leader trajectory have been developed in [48–51]. There are two typical types of feedback signals considered in the controller design of networked MASs, namely, *feedback signals without self-delay* and *feedback signals with self-delay* [52]. Effects of using different types of feedback are described as follows.

- **Feedback signals without self-delay**: Controllers designed from the feedback signals without self-delay are superior in maintaining stability, while the tracking performance degrades. The main reason is that the current states of the system are constantly forced to track the delayed signals. In other words, the tracking is less accurate when network delays increase.

- **Feedback signals with self-delay**: Using the feedback signals with self-delay in the controller effectively reduces the phase shift effect. The self-delay length can be either identical or different to the real network delays [52]. In contrast to the feedback signals without self-delay, controllers using feedback signals with self-delay help reduce the phase shift effect, but, however, make the system prone to being unstable, especially when the communication delays are long.

Motivated by different effects of the feedback signals, control policies proposed in [50] incorporate different types of feedback signals into a proportional controller with

---

[1] In this work, the term horizontal shift is used by assuming that the signals considered are periodic.

weighting coefficients, providing a trade-off between the tracking accuracy and the stability. However, the network delays tend to be time-varying, and the magnitude is not exactly known, resulting in the common use of feedback signals with inexact self-delays, such as the estimated delays [53], the average delays [54], or otherwise [55]. Then, the resultant errors caused by the discrepancy between the inexact self-delays and the actual delays can be treated as unknown but bounded disturbances [54].

Another control challenge in multi-manipulator systems is the complete pose (position and orientation) control. Kinematics and dynamics of the position and the orientation are highly coupled, giving rise to the difficulty of controlling the position and the orientation simultaneously. In control of the orientation, using the 3×3 rotation matrix is difficult to extract error vectors. The reason is that the nine entries are not independent but highly related by six constraints caused by the orthogonality condition [56]. Therefore, to facilitate the controllers design for the pose regulation, concepts of *minimal parameterizations* are adopted in the orientation calculations. Popular minimal parameterizations include Euler angle [57], angle/axis [58], and *unit quaternion* (also called *Euler parameters*) [15]. The minimal parameterizations have less parameters and constraints, and, thus, decrease the computation complexity in the expression of the orientation. The Euler angle and angle/axis have been broadly used in mobile robot control problems [59–61]. However, they may give rise to issues like singularities and discontinuity at certain points when mapping the rotation matrix to the Euler angle or the angle/axis. Fortunately, this drawback can be well dealt with using the unit quaternion representation that is singularity-free. Using the unit quaternion representation enables the decoupling control of the position and orientation, and, thus, facilitates the pose control design and its analysis [25, 26]. Although some attempts have been made for the full pose control of MASs using unit quaternion [24, 62], there remains a need for cooperative controllers that can regulate the full pose of a class of nonlinear multi-manipulator coordination systems in the presence of network delays.

In the leader-following MASs, active neighbour selection strategies have received considerable attention for tracking control tasks. The real-time and reliable network communications allow a robot to have multiple stable connections with its neighbours and/or the leader, which enhances the robot's situation awareness. In the majority of previous control approaches, all the transmitted data is immediately used in the control design. However, the quality (e.g., accuracy, noise level, etc.) of the neighbour signals may vary. Therefore, updating a well-performing agent's states based on the low-quality neighbour's information can impair the performance of this well-performing agent, and, furthermore, the overall MASs. For this reason, it would be helpful to consider a neighbour selection strategy to optimize the interaction between robots.

A great deal of research has studied the changing network connections in MASs (i.e., switching topology). However, they are essentially caused by the dynamical connections/disconnections of physical network transmissions over time. Network arrangement with such switching behaviour is referred to as *Passive Switching Topology* (PST). In many PST cases, efforts seek for the control stability under all possible topologies [63, 64]. Xie and Wang [65] proposed *Active/Strategic Switching Topology* (AST) by intentionally selecting two neighbours' states for each agent. This method, however, is more applicable for MASs with a large number of agents since it aims to reduce the communication complexity/cost [27, 65, 66]. Control approaches with AST for networked MASs with a small number of agents, such as multi-manipulator systems, have not been thoroughly studied. In addition, we have not encountered any work that encompasses the AST approach in a multi-manipulator system context to address the large tracking inaccuracy induced by the nonlinear state estimators.

2.3 Control of SMMS Systems

Although the autonomous multi-robotic systems could perform many high-level tasks, fully autonomous operation of the robot manipulators in unstructured environments is extremely difficult in the present day. Nevertheless, as human intervention and control can reduce the complexity of a purely autonomous robotic system, the concept of semi-autonomous multi-robotic systems has been attracting many researchers' interests. A marriage of the multi-robotic systems and teleoperation yields a system that combines the benefits of the two types of robotic systems. Some well-studied semi-autonomous frameworks with multiple robots are multiple-master-multiple-slave (MMMS) systems, multiple-master-single-slave (MMSS) systems, and SMMS systems. In this work, we are interested in SMMS manipulation systems.

SMMS systems are asymmetric teleoperation where one operator on the master side remotely commands multiple slave manipulators. Such systems have been applied to mobile teleoperation, such as mobile robot formation, remote navigation in uncertain, complex environments, and target searching and transportation. Many conventional control approaches have been used in the SMMS robotic systems. In [33, 67], a master manipulator remotely controls a formation of slave mobile ground vehicles using a feedback linearization approach and a PD control method. [68] proposes a teleoperated leader-follower formation of multiple unmanned aerial vehicles based on the artificial potential field. An adaptive semi-autonomous formation control for a group of differentially driven ground vehicles is presented in [69]. However, SMMS mobile robots with tasks such as formation only involve motion regulation. Nevertheless, it is essential for the SMMS telemanipulation applications to consider both the kinematic constraints and the force/torque constraints.

In cooperative manipulation tasks, the load distribution problem with the goal of reducing the internal force/torque needs to be thoroughly analyzed for better performance. Earlier results given in [70] address the load distribution problem by minimizing a linearly constrained quadratic function. This method has been broadly adopted in the research involving the load distribution [32]. However, as pointed out by the authors in [71], the load allocation strategy in [70] works for the manipulators with equality constraints. As a result, when the nonuniform load distribution is required by cooperative manipulators with heterogeneous payload capacities, additional torques may be generated and exerted on the manipulating object. This drawback motives the so-called non-squeezing load distribution method [72] to avoid internal stress. Furthermore, authors in [73] have shown that there exists no unique nonsqueezing load distribution solution and provided a more general load distribution paradigm. Based on the general load distribution solutions in [73], the work in [71] proposes an interesting dynamic load distribution approach for the multi-manipulator systems in the presence of dynamical payload constraints.

Like other networked robotic systems, SMMS manipulation systems are prone to instabilities due to various systematic nonlinearities and significant delays. One of the popular methods of ensuring stability is the time-domain passivity control scheme [74–76]. By investigating the energy content of input-output ports of a system, the artificial damping is added to dissipate the energy and, therefore, maintains the stability. However, the passivity-based control approaches impose passivity constraints on the human operator as well as the environment [77] that may not be satisfied in all applications. Model-based control methods, on the other hand, relax the strong limitations on the human and environments [78]. However, they require the exact information of dynamic and kinematic models. Therefore, controllers have been developed with adaptiveness and robustness to relieve the reliance on the precise information of the model and other undesired disturbances/frictions acting on robots [79–81].

Chapter 3

Background Theories

This chapter will introduce some prerequisite information of the following topics:

- Kinematic models of robotic manipulators
- Dynamic models of robotic manipulators
- Network communications
- Non-Singular Sliding Mode Control Theory.

These preliminaries and theories will be used to demonstrate the theoretical contributions and further developed in the following chapters.

3.1 Robotics Terminology

In this work, the term robot will mean an industrial manipulator, which is also commonly called a *robotic arm*. This type of robot is essentially a computer-controlled mechanical arm to implement human-like operating tasks. A robot manipulator consists of a sequence of *links* connected by *joints*. We consider a manipulator as a rigid body with all revolute joints that are like a hinge and allow relative rotation between links. We assume that each joint only has one *degree of freedom* (DOF) such that the joint number determines the DOF of a manipulator. The part of the robot connected to the ground is called the *base*, and the free end is called the *end effector*. *Task space* (or Cartesian space, operational space) is defined by the position and orientation of the end effector. *Joint space* is defined by a vector of the angular joint displacements. Also, to clarify the descriptions in the remaining text, the term position refers to a linear (or translational) position, while the angular position is referred to as orientation. Fig.3.1 shows a schematic of a two-link (2 DOFs) planar manipulator with the essential parts labelled.

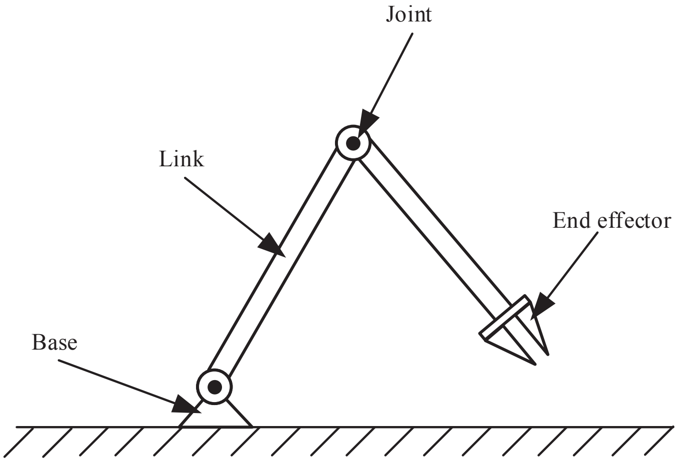


Figure 3.1: Schematic of a two-link planar robot manipulator illustrating the essential robotics terminology.

3.2 Kinematic Model

Manipulator kinematics studies the establishment of different coordinate systems as well as their mutual transformations to represent the positions and orientations of rigid manipulators. With the transformations, the pose (including position and orientation) of the end effector can be described as a function of joint variables with respect to the base frame, which is termed *forward kinematics*. One of the approaches to systematically derive the forward kinematics is by homogeneous transformations with the Denavit-Hartenberg (D-H) convention. In contrast, the *inverse kinematics* is to determine the joint variables that reproduce the given end effector trajectory. In this work, we do not require the computation of solutions for the inverse kinematics of motion states, since we assume the joint variables are directly measured.

3.2.1 Homogeneous Transformations

Homogeneous transformations provide a compact form to operate the rotation and translation through matrix multiplications. Consider two coordinate systems, denoted by $o_1x_1y_1z_1$ and $o_2x_2y_2z_2$, any point $\boldsymbol{p}$ has representations $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ with respect to frame $o_1x_1y_1z_1$ and frame $o_2x_2y_2z_2$, respectively. Then, relationship of the coordinates is established by:

$$\boldsymbol{p}_2 = \mathrm{T}_1^2 \boldsymbol{p}_1. \tag{3.1}$$

The transformation matrix, i.e., T_1^2, defines the homogeneous transformation from the frame $o_1x_1y_1z_1$ to the frame $o_2x_2y_2z_2$ given by

$$\mathrm{T}_1^2 = \begin{bmatrix} R_1^2 & \boldsymbol{d}_1^2 \\ \mathbf{0} & 1 \end{bmatrix} \in \mathrm{SE}(3), \tag{3.2}$$

where $R_1^2 \in \mathrm{SO}(3)$ [1] is the rotation matrix and $\boldsymbol{d}_1^2 \in \mathbb{R}^3$ denotes the translational distance from origin o_1 to origin o_2. Notice that the transformation defined by (3.2) is a special case as the last row is always taken to be [0 0 0 1]. Nevertheless, it has been proved to be practically useful for most of the manipulator systems [82].

More generally, for an open-chain manipulator constituted by $m + 1$ links and connected by m joints, the end effector pose with respect to the base frame can be determined by products of the homogeneous transformation matrices as

$$\mathrm{T}_e^b = \mathrm{T}_0^b \mathrm{T}_1^0 \mathrm{T}_2^1 ... \mathrm{T}_n^{m-1} \mathrm{T}_e^m. \tag{3.3}$$

In (3.2) as well as the recursive expression in (3.3), a series of frames should be predetermined for computing the coordinate transformations between them. In general, one can arbitrarily choose the frames as long as they are rigidly attached to the referring links. Unfortunately, in practice, many industrial manipulators have rather complex structures. Therefore, an arbitrary selection of frames may result in cumbersome expressions and make the control design and analysis very difficult. Nonetheless, by a clever choice of the origin and coordinate axes, we are able to reduce the number of parameters to formulate the homogeneous transformations. In the following, we present a convenient method for selecting frames, called D-H convention.

3.2.2 Denavit-Hartenberg Convention

In the D-H convention [82], the transformation between any two consecutive frames can be concisely expressed by four parameters, including the link length, link twist, link offset, and joint angle, denoted by a_{i-1}, α_{i-1}, d_i, and θ_i, accordingly. With reference to Fig. 3.2, definitions of the D-H parameters are given as:

---

[1] The notation SO(3) stands for Special Orthogonal group of order 3 and SE(3) stands for Special Euclidean group of order 3.

- a_i: Link length, distance from z_{i-1} to $z_{i'}$ measured along x_i.
- α_i: Link twist, angle between z_{i-1} and z_i about x_i to be taken positive when rotation is made counter-clockwise.
- d_i: Link offset, the distance from x_{i-1} to x_i measured along z_{i-1}.
- θ_i: Joint angle, the angle between x_{i-1} and x_i about z_{i-1} to be taken positive when rotation is made counter-clockwise.

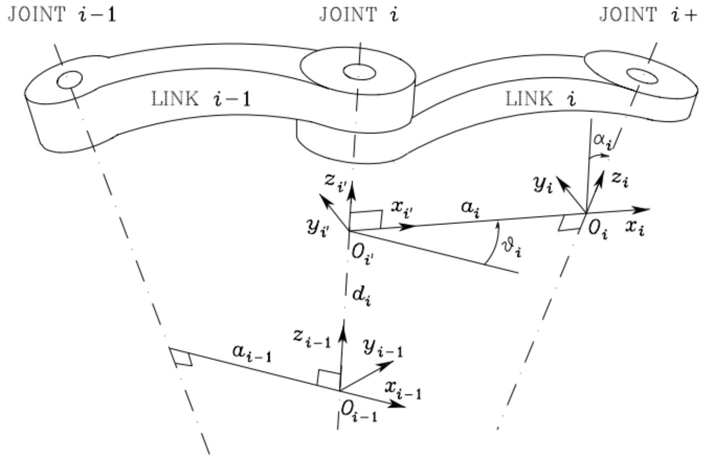


Figure 3.2: Coordinate frame assignment and D-H parameters for two links of a general manipulator [82].

The frame selection rules have been well described in [82]. To characterize the homogeneous transformations by the four parameters, the following two conditions should be satisfied, namely:

- The axis x_i is perpendicular to the axis z_{i-1}.
- The axis x_i intersects the axis z_{i-1}.

Under these conditions, there exists a unique homogeneous transformation along the serial rigid body, which can be generally expressed as:

$$\mathrm{T}_i^{i-1} = \begin{bmatrix} \mathrm{c}\theta_i & -\mathrm{s}\theta_i\mathrm{c}\alpha_i & \mathrm{s}\theta_i\mathrm{c}\alpha_i & a_i\mathrm{c}\theta_i \\ \mathrm{s}\theta_i & \mathrm{c}\theta_i\mathrm{c}\alpha_i & -\mathrm{c}\theta_i\mathrm{s}\alpha_i & a_i\mathrm{s}\theta_i \\ 0 & \mathrm{s}\alpha_i & \mathrm{c}\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.4}$$

where s and c denote the trigonometry of sin(·) and cos(·), respectively.

3.2.3 Jacobian Matrix

To further describe differential kinematics, a useful mapping matrix termed *geometric Jacobian* is introduced to relate the velocities and accelerations between the joint space and the task space. Since in this work we only consider manipulators with revolute joints, the Jacobian matrix can be computed by

$$J = \begin{bmatrix} J_p \\ J_o \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_{i-1} \times (\boldsymbol{p}_e - \boldsymbol{p}_{i-1}) \\ \boldsymbol{z}_{i-1} \end{bmatrix}, \tag{3.5}$$

where J_p and J_o are the Jacobian matrices for translational and angular operations, respectively. Using the transformation matrices as in (3.3) and (3.4), parameters to compute the Jacobian matrix are given as

$$z_i = R_1^0 R_1^2 \cdots R_{i-1}^{i-2} R_i^{i-1} z_0, \tag{3.6}$$

$$\tilde{\boldsymbol{p}}_e = \mathrm{T}_1^0 \mathrm{T}_1^2 \cdots \mathrm{T}_{n-1}^{n-2} \mathrm{T}_n^{n-1} \tilde{\boldsymbol{p}}_0, \tag{3.7}$$

$$\tilde{\boldsymbol{p}}_{i-1} = \mathrm{T}_1^0 \mathrm{T}_1^2 \cdots \mathrm{T}_{i-1}^{i-2} \tilde{\boldsymbol{p}}_0, \tag{3.8}$$

with $z_0 = [0\ \ 0\ \ 1]^T$, $\tilde{\boldsymbol{p}}_0 = [0\ 0\ 0\ 1]^T$, $\tilde{\boldsymbol{p}}_{i-1} = [\boldsymbol{p}_{i-1}^T\ \ 1]^T$, and $\tilde{\boldsymbol{p}}_e = [\boldsymbol{p}_e^T\ \ 1]^T$.

Then, the translational velocity (acceleration) and angular velocity (acceleration) of the end effector can be expressed as a function of the joint variables:

$$\begin{cases} \dot{\boldsymbol{x}} = J_p \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{x}} = \dot{J}_p \dot{\boldsymbol{q}} + J_p \ddot{\boldsymbol{q}}, \end{cases} \qquad \begin{cases} \boldsymbol{w} = J_o \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{w}} = \dot{J}_o \dot{\boldsymbol{q}} + J_o \ddot{\boldsymbol{q}}. \end{cases} \tag{3.9}$$

Unfortunately, in the orientation control, the commonly used 3×3 rotation matrix is difficult to extract error vectors. This is because the nine entries are not independent but highly related by six constraints caused by the orthogonality condition. To address this issue, the *minimal parameterization* methods have been utilized since they have fewer parameters and, thus, fewer constraints. As a result, the geometric Jacobian matrix may not be applicable. Instead, the analytical technique to compute the Jacobian matrices is considered. The geometric Jacobian and *analytical Jacobian* have the same form to determine the translational transformations, but, for the rotational computation, the geometric Jacobian J_o and the analytical Jacobian J_o^a are different yet related by

$$\begin{aligned} J_o(q) &= EJ_o^a(q), &(3.10)\\ J_o^a(q) &= E^{-1}J_o(q). &(3.11)\end{aligned}$$

In this form, the mapping matrix E is determined subject to the selection of minimal parameterization. For example, in this work, we consider the unit quaternion representation (denoted by ξ) and the E is given by

$$\begin{aligned} E &= 2H(\boldsymbol{\xi}), &(3.12)\\ E^{-1} &= \frac{1}{2}H^T(\boldsymbol{\xi}), &(3.13)\\ H(\boldsymbol{\xi}) &= \begin{bmatrix} -\boldsymbol{\xi}(2) & \boldsymbol{\xi}(1) & \boldsymbol{\xi}(4) & \boldsymbol{\xi}(3) \\ -\boldsymbol{\xi}(3) & \boldsymbol{\xi}(4) & \boldsymbol{\xi}(1) & -\boldsymbol{\xi}(2) \\ -\boldsymbol{\xi}(4) & -\boldsymbol{\xi}(3) & \boldsymbol{\xi}(2) & \boldsymbol{\xi}(1) \end{bmatrix}. &(3.14)\end{aligned}$$

From the human operators' viewpoint, it is intuitive to use task space control schemes, which are based on the task space deviation ($\Delta\boldsymbol{x}$) between the measured data and desired quantity. Then the control input generalized forces [2] ($\boldsymbol{u}$) has to be transformed into joint space generalized torque ($\boldsymbol{\tau}$) to reduce or to cancel the corresponding joint space deviation ($\Delta\boldsymbol{q}$), and in turn, to reduce or cancel $\Delta\boldsymbol{x}$. This can be realized through the *Jacobian transpose* as

$$\boldsymbol{\tau} = J^T\boldsymbol{u}. \qquad (3.15)$$

Notice that above equations hold as we assume that the considered manipulator is non-redundant [82, p. 167].

---

[2] In the remain text, task space generalized forces are called *forces*, while joint space generalized forces are termed *torques*.

3.2.4 Unit Quaternion

Unit quaternion is a 4-tuple, which can be denoted by $\boldsymbol{\xi} = [\eta\ \ \boldsymbol{\epsilon}^T]^T \in S^3$, where η is a scalar representing the real part of the quaternion, and $\boldsymbol{\epsilon} = [\epsilon_1\ \ \epsilon_2\ \ \epsilon_3]^T$ is a vector representing the imaginary part of the quaternion. With the D-H representation $\{\theta,\ a,\ d,\ \alpha\}$ provided in Page 62-63 in [82], the end effector orientation represented by the unit quaternion can be obtained by the forward kinematics in Lemma 3.1.

Lemma 3.1. *([25]) For a m-DOF manipulator, the pose of the k^{th} joint can be represented by the unit quaternion $\boldsymbol{\xi}_k$ and the extended position $\bar{\boldsymbol{p}}_k$ as*

$$\boldsymbol{\xi}_k = \begin{bmatrix} \cos(\theta_k/2)\cos(\alpha_k/2) \\ \cos(\theta_k/2)\sin(\alpha_k/2) \\ \sin(\theta_k/2)\sin(\alpha_k/2) \\ \sin(\theta_k/2)\cos(\alpha_k/2) \end{bmatrix}, \quad \bar{\boldsymbol{p}}_k = \begin{bmatrix} 0 \\ a_k\cos(\theta_k) \\ a_k\sin(\theta_k) \\ d_k \end{bmatrix}. \tag{3.16}$$

Then, the pose of the end effector $[\bar{\boldsymbol{p}}^T\ \ \boldsymbol{\xi}^T]^T$ can be calculated in the following form w.r.t. the base frame.

$$\begin{aligned} \boldsymbol{\xi}(\boldsymbol{q}) &= \boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_2 \otimes ... \otimes \boldsymbol{\xi}_k \otimes ... \otimes \boldsymbol{\xi}_{m-1} \otimes \boldsymbol{\xi}_m \\ \bar{\boldsymbol{p}}(\boldsymbol{q}) &= \boldsymbol{\xi}_1 \otimes ... \otimes \boldsymbol{\xi}_k \otimes ... \otimes \boldsymbol{\xi}_{m-1} \otimes \bar{\boldsymbol{p}}_m \otimes \boldsymbol{\xi}^*_{m-1} \otimes ... \otimes \boldsymbol{\xi}^*_1 \\ &\quad + \boldsymbol{\xi}_1 \otimes ... \otimes \boldsymbol{\xi}_{m-2} \otimes \bar{\boldsymbol{p}}_{m-1} \otimes \boldsymbol{\xi}^*_{m-2} \otimes ... \otimes \boldsymbol{\xi}^*_1 \\ &\quad + ... + \boldsymbol{\xi}_1 \otimes \bar{\boldsymbol{p}}_2 \otimes \boldsymbol{\xi}^*_1 + \bar{\boldsymbol{p}}_1. \end{aligned}$$

Given two unit quaternions $\boldsymbol{\xi}_1 = [\eta_1\ \ \boldsymbol{\epsilon}_1^T]^T$ and $\boldsymbol{\xi}_2 = [\eta_2\ \ \boldsymbol{\epsilon}_2^T]^T$, the addition acts component-wise, that is,

$$\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 = \begin{bmatrix} \eta_1 + \eta_2 \\ \boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_2 \end{bmatrix},$$

while the quaternion product $\otimes$ is defined in terms of the scalar and vector parts as

$$\boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_2 = \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\epsilon}_1^T\boldsymbol{\epsilon}_2 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + S(\boldsymbol{\epsilon}_1)\boldsymbol{\epsilon}_2 \end{bmatrix},$$

where $S(\boldsymbol{\epsilon})$ is a skew-symmetric matrix of $\boldsymbol{\epsilon} = [\epsilon_1\ \epsilon_2\ \epsilon_3]^{\mathrm{T}}$ defined as

$$S(\boldsymbol{\epsilon}) = \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & 0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & 0 \end{bmatrix}.$$

In the subsequent control design, the quaternion errors of angular position $\boldsymbol{e}_\xi$ and angular velocity $\dot{\boldsymbol{e}}_\xi$ are calculated in the sense of quaternion product given by

$$\boldsymbol{e}_\xi = \begin{bmatrix} e_\eta \\ \boldsymbol{e}_\epsilon \end{bmatrix} = \boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_2^* = \begin{bmatrix} \eta_1\eta_2 + \boldsymbol{\epsilon}_1^T \boldsymbol{\epsilon}_2 \\ -\eta_1 \boldsymbol{\epsilon}_2 + \eta_2 \boldsymbol{\epsilon}_1 - S(\boldsymbol{\epsilon}_1)\boldsymbol{\epsilon}_2 \end{bmatrix}, \tag{3.17}$$

where $\boldsymbol{\xi}_2^*$ is the conjugate of $\boldsymbol{\xi}_2$, i.e., $\boldsymbol{\xi}_2^* = [\eta_2 \quad -\boldsymbol{\epsilon}_2^T]^T$.

The angular velocity error is given by the time-derivative of quaternion error as

$$\dot{\boldsymbol{e}}_\xi = \begin{bmatrix} \dot{e}_\eta \\ \dot{\boldsymbol{e}}_\epsilon \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{e}_\epsilon^T \boldsymbol{e}_w \\ -\frac{1}{2}\Big(e_\eta I_3 + S(\boldsymbol{e}_\epsilon)\Big)\boldsymbol{e}_w - S(\boldsymbol{e}_\epsilon)\boldsymbol{w}_2 \end{bmatrix}, \tag{3.18}$$

where $\boldsymbol{e}_w = \boldsymbol{w}_2 - \boldsymbol{w}_1$, $\boldsymbol{w}_i \ (i = 1, 2)$ is the angular speed that can be calculated by

$$\boldsymbol{w}_i = 2 \begin{bmatrix} -\boldsymbol{\epsilon}_i^T \\ \eta_i I_3 - S(\boldsymbol{\epsilon}_i) \end{bmatrix}^T \dot{\boldsymbol{\xi}}_i. \tag{3.19}$$

The unit quaternion has several useful properties that will be used in the design and analysis of the control algorithm.

Property 3.1. *The scalar part and the imaginary part satisfy the quadratic norm constraint given by* $\eta^2 + \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = 1$.

Property 3.2. *The identity quaternion* $\boldsymbol{\xi}^I$ *with respect to the product is such that* $\boldsymbol{\xi}^I \otimes \boldsymbol{\xi}_1 = \boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}^I = \boldsymbol{\xi}_1$. *It encodes the null rotation and can be expressed in a quaternion form as* $\boldsymbol{\xi}^I = [1 \ \ \mathbf{0}^T]^T$.

Property 3.3. *The norm of a unit quaternion satisfies* $\|\boldsymbol{\xi}\| = 1$ *and the quaternion product of two unit quaternions has the property of* $\|\boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_2\| = \|\boldsymbol{\xi}_2 \otimes \boldsymbol{\xi}_1\| = \|\boldsymbol{\xi}_1\|\|\boldsymbol{\xi}_2\|$.

Property 3.4. *The quaternion product is associative, e.g.,* $(\boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_2) \otimes \boldsymbol{\xi}_3 = \boldsymbol{\xi}_1 \otimes (\boldsymbol{\xi}_2 \otimes \boldsymbol{\xi}_3)$, *and distributive over the sum, e.g.,* $(\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2) \otimes \boldsymbol{\xi}_3 = \boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_3 + \boldsymbol{\xi}_2 \otimes \boldsymbol{\xi}_3$.

Property 3.5. *The relationship between the unit quaternion and the rotation matrix* $R \in \mathbb{R}^{3\times3}$ *is given by*

$$R(\eta, \boldsymbol{\epsilon}) = (\eta^2 + \boldsymbol{\epsilon}^T\boldsymbol{\epsilon})I + 2\eta S(\boldsymbol{\epsilon}) + 3\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T$$

where $S(*)$ *is a skew-symmetric matrix.*

Lemma 3.2. *([25]) Consider the fact that a quaternion and its negative encode the same rotation, one orientation represented by* $\boldsymbol{\xi}_1$ *is aligned with another one represented by* $\boldsymbol{\xi}_2$, *either when* $\boldsymbol{\xi}_1 = \boldsymbol{\xi}_2$, *resulting in the quaternion error as* $\boldsymbol{e}_\xi = \boldsymbol{\xi}_1 \otimes \boldsymbol{\xi}_2^* = [1 \ \ \mathbf{0}^T]^T = \boldsymbol{\xi}^I$, *or when* $\boldsymbol{\xi}_1 = -\boldsymbol{\xi}_2$, *resulting in* $\boldsymbol{e}_\xi = [-1 \ \ \mathbf{0}^T]^T = -\boldsymbol{\xi}^I$.

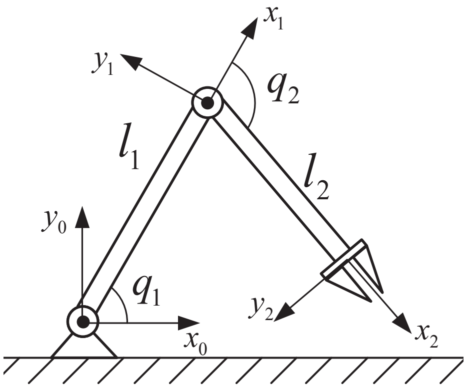


Figure 3.3: Coordinate frame assignment of a two-link planar manipulator.

3.2.5 Examples

Based on the aforementioned concepts, in the following, we present two types of manipulators that will be used to illustrate the proposed theories in the subsequent chapters.

Two-link Planar Manipulator

Consider a typical two-link planar robot arm shown in Fig.3.3, each frame is established with the x-axis along the relative link direction. Such frame selection results in a simple set of D-H parameters, as specified in Table. 3.1.

Table 3.1: D-H parameters for 2-DOF planar manipulator.

Link	a_i	α_i	d_i	θ_i
1	l_1	0	0	q_1
2	l_2	0	0	q_2

As the homogeneous transformation has the same structure for each joint, we obtain the forward kinematics function as

$$T_2^0(\boldsymbol{q}_i) = \begin{bmatrix} \mathrm{c}_{12} & -\mathrm{s}_{12} & 0 & l_1\mathrm{c}_1 + l_2\mathrm{c}_{12} \\ \mathrm{s}_{12} & \mathrm{c}_{12} & 0 & l_1\mathrm{s}_1 + l_2\mathrm{s}_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.20}$$

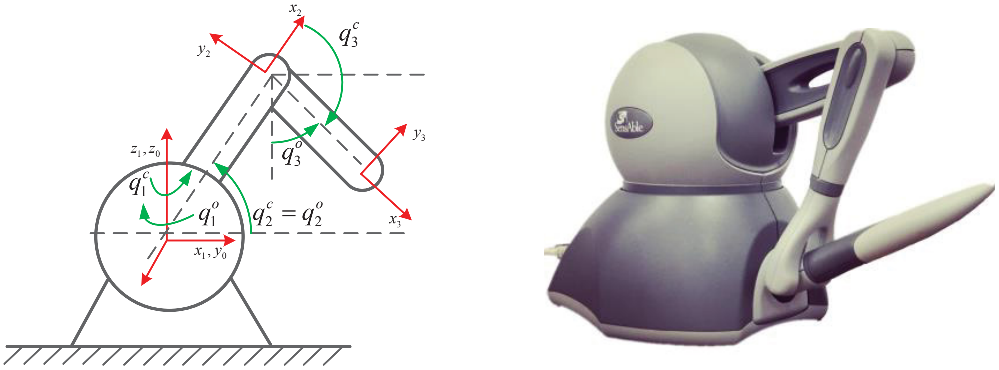
where $\mathrm{c}_1 = \cos(q_1)$, $\mathrm{s}_1 = \sin(q_1)$, $\mathrm{c}_{12} = \cos(q_1 + q_2)$, $\mathrm{s}_{12} = \sin(q_1 + q_2)$.

For the differential kinematics, the geometric Jacobian matrix is given by

$$J = \begin{bmatrix} J_p \\ J_o \end{bmatrix} = \begin{bmatrix} -l_1 \mathrm{s}_1 - l_2 \mathrm{s}_{12} & -l_2 \mathrm{s}_{12} \\ l_1 \mathrm{c}_1 + l_2 \mathrm{c}_{12} & l_2 \mathrm{c}_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \tag{3.21}$$

where $J_p \in \mathbb{R}^{3\times 2}$ and $J_o \in \mathbb{R}^{3\times 2}$ are the translational Jacobian matrix and angular Jacobian matrix, respectively.

Three-DOF Phantom Omni haptic device



(a) Coordinate frame assignment of an Phantom Omni haptic device.

(b) Picture of an Sensable Phantom Omni haptic device.

Figure 3.4: Schematics and picture of the Phantom Omni haptic device.

In the numerical and experimental studies, the 3-DOF Phantom Omni haptic devices[3] were also used to implement the proposed theory. The haptic device has 3 actuated revolute joints and its first link length is 0, i.e., $l_1 = 0$. The frames are determined as depicted in Fig.3.4 (a) and the resultant D-H parameters are shown in Table.3.2. As a result, the pose of the end effector can be obtained from the

---

[3] https://www.3dsystems.com

Table 3.2: D-H parameters of the Phantom Omni device [83].

Link	a_i	α_i	d_i	θ_i
1	0	0	l_1	q_1
2	l_2	0	0	q_2
3	l_2	0	0	q_3

homogeneous transformation matrix as

$$T_3^0(\boldsymbol{q}_i) = \begin{bmatrix} \mathrm{c}_1\mathrm{c}_{23} & -\mathrm{c}_1\mathrm{s}_{23} & \mathrm{s}_1 & r\mathrm{c}_1 \\ \mathrm{s}_1\mathrm{c}_{23} & -\mathrm{s}_1\mathrm{s}_{23} & -\mathrm{c}_1 & r\mathrm{s}_1 \\ \mathrm{s}_{23} & \mathrm{c}_{23} & 0 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.22}$$

where $r = l_2\mathrm{c}_2 + l_3\mathrm{c}_{23}$ and $h = l_2\mathrm{s}_2 + l_3\mathrm{s}_{23}$.

The expression of the geometric Jacobian matrix is given by

$$J = \begin{bmatrix} J_p \\ J_o \end{bmatrix} = \begin{bmatrix} -r\mathrm{s}_1 & -h\mathrm{c}_1 & -l_3\mathrm{c}_1\mathrm{s}_{23} \\ r\mathrm{c}_1 & -h\mathrm{s}_1 & -l_3\mathrm{s}_1\mathrm{s}_{23} \\ 0 & r & l_3\mathrm{c}_{23} \\ 0 & 0 & \mathrm{s}_1 \\ 0 & 0 & -\mathrm{c}_1 \\ 1 & 0 & 0 \end{bmatrix}. \tag{3.23}$$

In physical experiments, the measured joint angles from the Phantom Omni QUARC block do not match the conventions used in the kinematic model of the device, so there exists a conversion between the conventional angles $\boldsymbol{q}_i^{\mathrm{c}}$ and measured angles $\boldsymbol{q}^{\mathrm{o}} = [q_1^{\mathrm{o}}\ q_2^{\mathrm{o}}\ q_3^{\mathrm{o}}]^T$ as

$$\boldsymbol{q}^{\mathrm{c}} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \boldsymbol{q}^{\mathrm{o}} - \begin{bmatrix} 0 \\ 0 \\ \pi/2 \end{bmatrix}. \tag{3.24}$$

3.3 Dynamic Model

Dynamic equations of a manipulator are studied to explicitly describe the relationship between force and motion. In addition, it allows us to simulate the manipulator motion and control algorithms without having a physically available system.

In this work, we use the E-L equations to formulate the dynamic model. It has been proved to be practically useful in representing nonlinear mechanical systems. In addition, it is computationally efficient and has several facilitating properties in particular for developing control algorithms, such as globally bounded inertia matrix, skew symmetry property, etc. Derivations of the E-L equations have been well discussed in many existing literature [15, 84]. In what follows, we describe the general forms of the E-L representations in both joint space and task space.

3.3.1 Joint Space Dynamics

For many robot arms, the default measurements are taken in joint space, e.g. encoder measurements. Therefore, from the measurement viewpoint, it is straightforward to design a controller with the joint-space dynamics. Consider a m DOFs robot arm (indexed by i) in a multi-manipulator system, the mathematical model in joint space is assumed to have the following E-L form:

$$\mathbb{M}_i(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \mathbb{C}_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \mathbb{G}_i(\boldsymbol{q}_i) + \boldsymbol{\tau}_i^n = \boldsymbol{\tau}_i - \boldsymbol{\tau}_i^e, \tag{3.25}$$

where $\boldsymbol{q}_i \in \mathbb{R}^m$, $\dot{\boldsymbol{q}}_i \in \mathbb{R}^m$, and $\ddot{\boldsymbol{q}}_i \in \mathbb{R}^m$ are the angular position, velocity and acceleration. The matrices and vector, $\mathbb{M}_i$, $\mathbb{C}_i$, and $\mathbb{G}_i$, contain structure parameters of the manipulator, such as link length and joint mass. Their definitions are given as: $\mathbb{M}_i \in \mathbb{R}^{m\times m}$ is the symmetric and uniformly positive definite inertia matrix. $\mathbb{C}_i \in \mathbb{R}^{m\times m}$ is the Coriolis and centrifugal loading matrix. $\mathbb{G}_i \in \mathbb{R}^m$ is the gravitational loading vector. $\boldsymbol{\tau}_i \in \mathbb{R}^m$ denotes the designed torque control input and $\boldsymbol{\tau}_i^e \in \mathbb{R}^m$ represents the equivalent environment torque exerted on the manipulator. $\boldsymbol{\tau}_i^e = \boldsymbol{0}$ if only the free motion regulation is considered. $\boldsymbol{\tau}_i^{\mathrm{n}}$ consists of the disturbance and the friction, denoted by $\boldsymbol{\tau}_i^{\mathrm{d}} \in \mathbb{R}^m$ and $\boldsymbol{\tau}_i^{\mathrm{f}} \in \mathbb{R}^m$, respectively. In other words, $\boldsymbol{\tau}_i^{\mathrm{n}} = \boldsymbol{\tau}_i^{\mathrm{f}} + \boldsymbol{\tau}_i^{\mathrm{d}}$. The configuration dependant friction model is given as $\boldsymbol{\tau}_i^{\mathrm{f}} = \gamma_i\dot{\boldsymbol{q}}_i$, where γ_i is an unknown coefficient.

3.3.2 Task Space Dynamics

In spite of the convenience of using the dynamic model in joint space, it is more intuitive to control the motion of the end effector where the tasks are to be specified to the end effector. By using the forward kinematics, it is easy to acquire indirect measurements in task space. As well, in some applications where measurements are implemented in task space, such as visual measurements, using the task space kinematic model is more efficient.

Like the joint space dynamic model in (3.25), the mathematical dynamic model in task space can be formulated by the E-L representation as

$$M_i(\boldsymbol{q}_i)\ddot{\boldsymbol{x}}_i + C_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{x}}_i + \boldsymbol{g}_i(\boldsymbol{q}_i) + \boldsymbol{f}_i^{\mathrm{n}}(\dot{\boldsymbol{q}}_i) = \boldsymbol{u}_i - \boldsymbol{f}_i^{\mathrm{e}}, \tag{3.26}$$

where $\boldsymbol{x}_i \in \mathbb{R}^3$, $\dot{\boldsymbol{x}}_i \in \mathbb{R}^3$, and $\ddot{\boldsymbol{x}}_i \in \mathbb{R}^3$ are position, velocity, and acceleration of the end effector. $M_i(\boldsymbol{q}_i) \in \mathbb{R}^{3\times3}$, $C_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) \in \mathbb{R}^{3\times3}$, and $\boldsymbol{g}_i(\boldsymbol{q}_i) \in \mathbb{R}^3$ are systematic parameters that have the same definitions as those in (3.25). $\boldsymbol{u}_i \in \mathbb{R}^3$ denotes the control signals and $\boldsymbol{f}_i^{\mathrm{e}} \in \mathbb{R}^3$ is the environmental force. The disturbance and friction in task space are denoted by $\boldsymbol{d}_i \in \mathbb{R}^3$ and $\boldsymbol{f}_i^{\mathrm{f}} \in \mathbb{R}^3$, respectively, and $\boldsymbol{f}_i^{\mathrm{n}} = \boldsymbol{f}_i^{\mathrm{f}} + \boldsymbol{d}_i$.

The friction model in this paper takes into account the effects of the Coulomb friction, static friction, viscous friction and Stribeck effect [85], and is given as a function of the angular velocity $\dot{\boldsymbol{q}}_i$ as

$$\boldsymbol{f}_i^f = J_i\Big\{\gamma_i^1[tanh(\gamma_i^2\dot{\boldsymbol{q}}_i) - tanh(\gamma_i^3\dot{\boldsymbol{q}}_i)] + \gamma_i^4 tanh(\gamma_i^5\dot{\boldsymbol{q}}_i) + \gamma_i^6\dot{\boldsymbol{q}}_i\Big\}, \tag{3.27}$$

where $\gamma_i^p(p = 1, 2, \ldots, 6)$ are unknown coefficients.

Assume that the manipulator model is non-redundant and in a non-singular configuration, the relationship between the dynamic models in (3.25) and (3.26) is established as follows.

Property 3.6. *The Jacobian matrix defined by* $J_i = \partial\boldsymbol{x}_i/\partial\boldsymbol{q}_i \in \mathbb{R}^{3\times m}$ *encodes relationships between velocities as* $\dot{\boldsymbol{x}}_i = J_i\dot{\boldsymbol{q}}_i$.

Property 3.7. *The Jacobian matrix also describes the linear transformations of the force variables, namely,* $\boldsymbol{f}_i^f = J_i\boldsymbol{\tau}_i^f$, $\boldsymbol{d}_i = J_i\boldsymbol{\tau}_i^d$, $\boldsymbol{f}_i^f = J_i\boldsymbol{\tau}_i^f$, $\boldsymbol{f}_i^e = J_i\boldsymbol{\tau}_i^e$, *and* $\boldsymbol{u}_i = J_i\boldsymbol{\tau}_i$.

Property 3.8. *The relationship between the dynamics matrices/vectors in joint space and those in task space can be described by* $M_i = J_i^{-T}\mathbb{M}_i J_i^{-1}$, $C_i = J_i^{-T}\mathbb{C}_i J_i^{-1} - J_i^{-T}\mathbb{M}_i J_i^{-1}\dot{J}_i J_i^{-1}$, *and* $\boldsymbol{g}_i = J_i^{-T}\mathbb{G}_i$ *[86].*

In practice, due to the material properties, measurement variability or thermal effect, the dynamic model usually contains a certain amount of uncertainties. Taking the dynamic model in (3.26) as an example, the actual parametric matrices/vectors, M_i, C_i, and $\boldsymbol{g}_i$, can be specified as a known nominal portion plus an unknown yet bounded variation from the nominal values, that is, $M_i = \bar{M}_i + \Delta M_i$, $C_i = \bar{C}_i + \Delta C_i$, and $\boldsymbol{g}_i = \bar{\boldsymbol{g}}_i + \Delta \boldsymbol{g}_i$, where $\bar{M}_i$,$\bar{C}_i$, and $\bar{\boldsymbol{g}}_i$ are nominal matrices/vector. ΔM_i, ΔC_i, and $\Delta \boldsymbol{g}_i$ are unknown modeling uncertainties. Then, the model (3.26) can be rewritten as

$$\bar{M}_i \ddot{\boldsymbol{x}}_i + \bar{C}_i \dot{\boldsymbol{x}}_i + \bar{\boldsymbol{g}}_i = \boldsymbol{u}_i + \boldsymbol{\rho}_i. \tag{3.28}$$

Assumption 3.1. *[43] $\boldsymbol{\rho}_i$ comprises all of the unknown parameters, i.e., $\boldsymbol{\rho}_i = \boldsymbol{f}_i^n - (\Delta M_i \ddot{\boldsymbol{x}}_i + \Delta C_i \dot{\boldsymbol{x}}_i + \Delta \boldsymbol{g}_i)$. By assuming the following boundedness: $\|\Delta M_i\|_1 \leq \mathbb{K}_1$, $\|\Delta C_i \boldsymbol{x}_i + \Delta \boldsymbol{g}_i\|_1 \leq \mathbb{K}_2 + \mathbb{K}_3 \|\boldsymbol{x}_i\|_1 + \mathbb{K}_4 \|\dot{\boldsymbol{x}}_i\|_1^2$, $\|\boldsymbol{d}_i\|_1 \leq \mathbb{K}_5$, and $\|\boldsymbol{f}_i^f\|_1 \leq \mathbb{K}_6 \|\dot{\boldsymbol{x}}_i\|_1$, we obtain*

$$\|\boldsymbol{\rho}_i\|_1 \quad \leq \quad \Theta_i \Phi_i, \tag{3.29}$$

where Θ_i represents the unknown parameters consisting of $\mathbb{K}_1 \sim \mathbb{K}_6$ that determine the parameter bounds. $\Phi_i = 1 + \|\boldsymbol{x}_i\|_1 + \|\dot{\boldsymbol{x}}_i\|_1^2 > 0$. $\| * \|_1$ denotes the L-1 norm.

3.3.3 Examples

In what follows, we present the parametric matrices and vectors of the dynamic models in joint space of the two types of manipulators, as shown in Fig.3.3 and Fig.3.4 (a).

Two-link planar manipulator

The parametric matrices and vector in joint space of a 2-DOF planar manipulator is given by

$$\mathbb{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \ \mathbb{G} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \ \mathbb{C} = \begin{bmatrix} -c\dot{q}_2 & -c(\dot{q}_1 + \dot{q}_2) \\ c\dot{q}_1 & 0 \end{bmatrix},$$

where

$M_{11} = m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} \cos(q_2)) + I_1 + I_2,$

$M_{12} = M_{21} = m_2(l_{c2}^2 + l_1 l_{c2} \cos(q_2)) + I_2,$

$M_{22} = m_2 l_{c2}^2 + I_2,$

$c = m_2 l_1 l_{c2} \sin(q_2),$

$g1 = (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2),$

$g2 = m_2 l_{c2} g \cos(q_1 + q_2)$.

Definitions of the relevant variables are given as m_1 and m_2 are link mass, l_1 and l_2 are link length, l_{c1} and l_{c2} are the distances from the joint to the centre of mass of the links, I_1 and I_2 are the moments of inertia with respect to the centres of mass of the two links, and g is the gravity acceleration.

In the subsequent chapters, when the two-link planar robot arm is used, the following data [78] is assigned with proper units: $l_1 = 1$, $l_2 = 0.8$, $l_{c1} = 0.5$, $l_{c2} = 0.5$, $I_1 = 5$, $I_2 = 5$, $m_1 = 0.5$, $m_2 = 1.5$, $g = 9.8\ m/s^2$.

Three-DOF Phantom Omni haptic device

The parametric matrices and vector in joint space of a three-DOF Phantom Omni haptic device is given by

$$\mathbb{M} = \begin{bmatrix} h_{11} & 0 & 0 \\ 0 & h_{22} & h_{23} \\ 0 & h_{32} & h_{33} \end{bmatrix}, \ \mathbb{G} = \begin{bmatrix} 0 \\ \theta_5 g \mathrm{c}_2 + \theta_6 g \mathrm{c}_{23} \\ \theta_6 g \mathrm{c}_{23} \end{bmatrix},$$

$$\mathbb{C} = \begin{bmatrix} -(a_1\dot{q}_2 + a_2\dot{q}_3) & -a_1\dot{q}_1 & -a_2\dot{q}_1 \\ a_1\dot{q}_1 & -a_3\dot{q}_3 & -a_3(\dot{q}_2 + \dot{q}_3) \\ a_2\dot{q}_1 & a_3\dot{q}_2 & 0 \end{bmatrix},$$

where

$h_{11} = \theta_1 + \theta_2 \mathrm{c}_2^2 + \theta_3 \mathrm{c}_{23}^2 + 2\theta_4 \mathrm{c}_2 \mathrm{c}_{23}$

$h_{22} = \theta_2 + \theta_3 + 2\theta_4 \mathrm{c}_3$

$h_{23} = \theta_3 + \theta_4 \mathrm{c}_3$

$h_{33} = \theta_3$,

$a_1 = \theta_2 \mathrm{c}_2 s_2 + \theta_3 \mathrm{c}_{23} \mathrm{s}_{23} + \theta_4 \mathrm{c}_{2*23}$

$a_2 = \theta_3 \mathrm{c}_{23} \mathrm{s}_{23} + \theta_4 \mathrm{c}_2 \mathrm{s}_{23}$,

$a_3 = \theta_4 \mathrm{s}_3$

with $\mathrm{s}_2 = \sin(q_2)$, $\mathrm{c}_2 = \cos(q_2)$, $\mathrm{s}_3 = \sin(q_3)$, $\mathrm{c}_3 = \cos(q_3)$, $\mathrm{s}_{23} = \sin(q_2 + q_3)$, $\mathrm{c}_{23} = \cos(q_2 + q_3)$, and $\mathrm{c}_{2*23} = \cos(2q_2 + q_3)$. Estimated values to compute $\mathbb{M}$, $\mathbb{C}$, and $\mathbb{G}$ are given in Table 3.3. Some mechanical properties to describe the limits of the haptic device are listed in Table.3.4 [83].

Table 3.3: Dynamic parameters of the Phantom Omni device [83]

Parameter	Value	Parameter	Value
$l_1(m)$	0	$\theta_2(kg \cdot m^2)$	7.0×10^{-3}
$l_2(m)$	0.135	$\theta_3(kg \cdot m^2)$	8.0×10^{-3}
$l_3(m)$	0.130	$\theta_4(kg \cdot m^2)$	0.4×10^{-3}
$g(N/kg)$	9.8	$\theta_5(kg \cdot m)$	9.1×10^{-3}
$\theta_1(kg \cdot m^2)$	3.7×10^{-3}	$\theta_6(kg \cdot m)$	5.2×10^{-3}

Table 3.4: Mechanical limits of the Phantom Omni device

Joint	Limits		
	Position(rad)	Torque (Nm)	Static Friction (Nm)
1	$[-\frac{\pi}{3},\ \frac{\pi}{3}]$	0.30	N/A
2	$[0,\ 1.79]$	0.29	0.065
3	$[-2.45,\ -0.25]$	0.20	0.028

3.4 Network Communications

3.4.1 Network Representation

Consider a network with a leader and a finite number of follower agents, the connection and information flow between agents can be described by a directed graph expressed as

$$G = (\nu, \varepsilon, A, b), \tag{3.30}$$

where agents (or nodes) are indexed by ν. ε denotes the edge, and the adjacency matrix $A = [a_{ij}]$ describes the connection between agents where $a_{ij} > 0$ if $(i, j) \in \varepsilon$ and $a_{ij} = 0$ otherwise. The connection between the follower agent and the leader is represented as $b = [b_1, b_2, ..., b_n]$ where $b_i > 0$ if the i^{th} agent is connected to the leader and $b_i = 0$ otherwise.

3.4.2 Directed Graph Connectivity

A graph is called *directed graph* when its edges have a direction associated with them. In the networked control system, the graph connectivity has a significant effect on the performance and stability. The definitions of some examples of connectivity are given as follows and the corresponding graphs are depicted in Fig.3.5.

A digraph is *strongly connected* if from each node to each other node there is a directed walk (See Fig.3.5 (a)), while a digraph is *weakly connected* if such a directed

(a) Strongly connected (b) Weakly connected

(c) Balanced graph (d) Unbalanced graph

Figure 3.5: Examples of different types of directed graph connectivity.

walk does not always exist for at least one node, but the underlying graph is connected (See Fig.3.5 (b)).In a directed graph, the node is balanced if and only if its in-degree and out-degree are equal and a graph is called *balanced* if and only if all of its nodes are balanced (See Fig.3.5 (c)). Otherwise, the node and the graph are *unbalanced* (See Fig.3.5 (d)).

3.4.3 Network-Induced Constraints

In this work, we consider a weakly connected and unbalanced directed graph with one leader and a finite number of followers. This makes our theoretical studies more challenging yet more practical. In addition, we assume that the leader is globally reachable to the followers, that is, from the leader to each follower agent, there always exists a directed path [87].

In the following, we specify two network communication constraints, namely, the network delays and the packet loss.

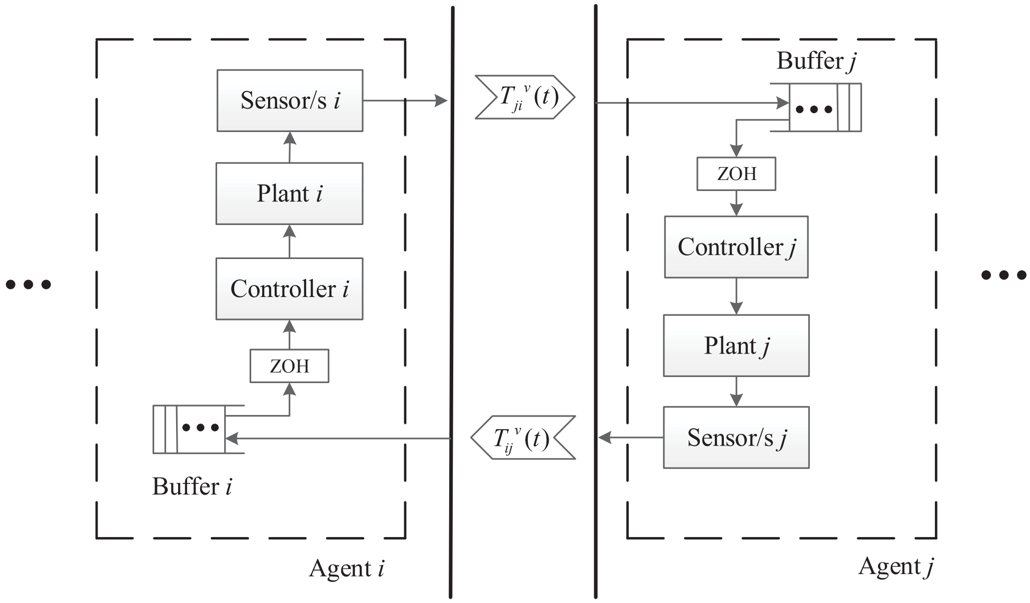


Figure 3.6: Schematic of a network control system of two agents with time-varying delays and the packet loss.

Time-Varying Delays

Based on the physical test in [88] and [89], the time-varying network delays $T_{ij}^{\mathrm{v}}(t)$ from agent j to i can be expressed as

$$T_{ij}^{\mathrm{v}}(t) = \bar{T}_{ij}^{\mathrm{v}} + \delta_{ij}^{\mathrm{v}}(t), \tag{3.31}$$

where $\bar{T}_{ij}^{\mathrm{v}}$ is a known and constant nominal value and it is assumed that each channel experiences the same average delay. $\delta_{ij}^{\mathrm{v}}(t)$ is the unknown bounded deviation.

Packet Loss

The packet loss probability can be expressed by the Gilbert-Elliott model with the classical 2-state ($\mathbb{G}$ denotes the good state and $\mathbb{B}$ denotes the bad state) Markov model [90, 91]. Let $\mathbb{S}_{ij}(t_k)$ denote the network states from agent j to i at time t_k. Then the probability of state transition can be represented by

$$\begin{aligned} P_{\mathbb{G}\to\mathbb{B}} &= P(\mathbb{S}_{ij}(t_{k+1}) = \mathbb{B} | \mathbb{S}_{ij}(t_k) = \mathbb{G}) \\ P_{\mathbb{B}\to\mathbb{G}} &= P(\mathbb{S}_{ij}(t_{k+1}) = \mathbb{G} | \mathbb{S}_{ij}(t_k) = \mathbb{B}). \end{aligned}$$

In the receiver, a buffer is used to store all the successfully transmitted data, as shown in Fig.3.6. Taking the position signals as an example, the neighbour information that feeds into the receiver's controller is expressed by the Simple Gilbert Model (SGM) as

$$\boldsymbol{x}_j(T_{ij}(t_k)) = (1-\theta)\boldsymbol{x}_j(T_{ij}^{\mathrm{v}}(t_k)) + \theta\boldsymbol{x}_j(T_{ij}^{\mathrm{v}}(t_{k-1})), \tag{3.32}$$

where $\theta = 0$ if $\boldsymbol{x}_j(T_{ij}^{\mathrm{v}}(t_k))$ is transmitted successfully and $\theta = 1$ otherwise. To capture bursty behaviour in packet loss, we define

$$\boldsymbol{x}_j(T_{ij}^{\mathrm{v}}(t_{k-1})) = \boldsymbol{x}_j(T_{ij}^{\mathrm{v}}(t_{k-2})) = \cdots = \boldsymbol{x}_j(T_{ij}^{\mathrm{v}}(t_{k-\ell})), \tag{3.33}$$

where ℓ is the bursty length.

From (3.32) and (3.33), it is equivalent to regard the packet loss as additional time delays. Therefore, by denoting $\Delta_t = t_k - t_{k-\ell}$, the equivalent delay, $T_{ij}(t_k)$, becomes

$$T_{ij}(t_k) = (1-\theta)T_{ij}^{\mathrm{v}}(t_k) + \theta(T_{ij}^{\mathrm{v}}(t_{k-\ell}) + \Delta_t). \tag{3.34}$$

Experimental Setting

In our experiments, the control software used to establish network communications is provided by Quanser QUARC 2.6[4]. QUARC has a high-level communication blockset (*Stream Client* and *Stream Server*) to establish the connections between the server host and the client host. Packets are sent/received between two personal computers (PCs) connected to the Local Area Network (LAN) at the Advance Control and Mechatronics Lab, Dalhousie University, Canada. Packets are transmitted by using User Datagram Protocol (UDP) at an updating frequency of 1000 Hz. Packets are sized 72 bytes in most of our tests. Generally, LAN can be categorized into *Wired (Ethernet) LAN* and *Wireless LAN*. Fig.3.7 shows the measured one-way network delays and its normal distribution by using a wired LAN and wireless LAN, respectively. Wired LAN transmits electric signals that flow over the cables with less interference. Thus, the data transmission experiences less and constant delays, as illustrated in Fig.3.7 (a) and (b) where the nominal delay was $\bar{T} = 1.4 \times 10^{-3}$ *sec* with a standard error of $\sigma = 0.00026$ *sec*. Wireless LAN enables robots to move around in a broader

---

[4] https://www.quanser.com/products/quarc-real-time-control-software/

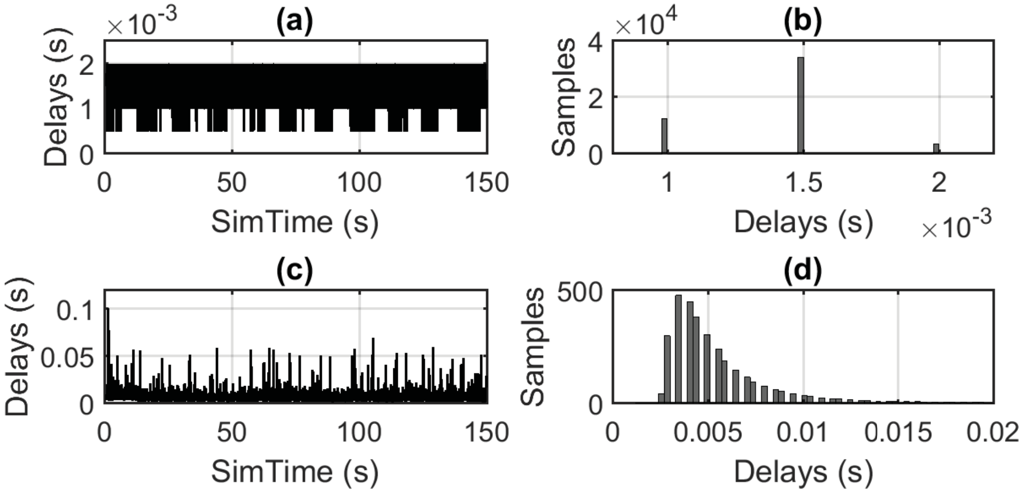


Figure 3.7: (a) Measured one-way network delays and (b) its normal distribution in a wired LAN; (c) Measured one-way network delays and (d) its normal distribution in a wireless LAN.

range, but Wireless LAN appears to suffer from interference of various types during travel from the source to the destination. The WiFi cards used in the test are Tenda W311M Nano USB Adapter(Wireless N150 speed up to 150 *Mbps*). As shown in Fig.3.7 (c) and (d), significant *jitters* are presented, and the nominal delay increases to $\bar{T} = 6.3 \times 10^{-3}$ *sec* with $\sigma = 0.0057$ *sec*.

3.5 Non-Singular Terminal Sliding Mode Control Theory

To improve the robustness of manipulation systems, the SMC method is studied to make the system less sensitive to the external noises and other unknown disturbance sources. In addition, the NTSM [42, 43] was developed so that the sliding surface as well as the zero errors can be realized in a finite time and meanwhile avoid singularities. The basis of the NTSM algorithm is introduced as follows. Consider a nonlinear dynamic system in a general form as

$$\begin{cases} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= f(\boldsymbol{x}) + b(\boldsymbol{x})u + d(\boldsymbol{x}), \end{cases} \tag{3.35}$$

where $\boldsymbol{x} = [x_1 \ \ x_2]^T$ is the system state, $f(\boldsymbol{x})$ and $b(\boldsymbol{x})$ are smooth nonlinear functions, and $d(\boldsymbol{x})$ is the external disturbances with a known bound, i.e., $|d(x)| \leq D$ and $D \geq 0$.

To achieve $x_1 \to 0$ and $x_2 \to 0$ as $t \to t_c$ where t_c is a finite time, a sliding manifold s is constructed as

$$s = x_1 + \beta x_2^{\alpha}, \tag{3.36}$$

where $\beta > 0$. $\alpha = (2h+1)/(2h-1)$ with $h = 1, 2, ...$, such that $1 < \alpha < 2$ [43]. The NTSM controller u designed from the sliding surface s enables the finite-time convergence and disturbance insensitiveness. One standard controller [43] for the system in (3.35) is designed as:

$$u = -b^{-1}(\boldsymbol{x})[f(\boldsymbol{x}) + \frac{x_2^{(2-\alpha)}}{\alpha\beta} + (D+\kappa)\mathrm{sgn}(s)], \tag{3.37}$$

where $\kappa > 0$ and sgn$(\cdot)$ denotes a sign function, that is,

$$\mathrm{sgn}(a) = \begin{cases} 1, & \text{if } a > 0 \\ 0, & \text{if } a = 0 \\ -1, & \text{if } a < 0. \end{cases} \tag{3.38}$$

In the finite-time stability analysis using the Lyapunov method, the following lemma gives the criteria for determining the convergence time.

Lemma 3.3. *[92] For a non-Lipschitz continuous non-linear system $\dot{x} = f(x)$, suppose there exists a continuous function $V(x)$ defined on a neighbourhood of the origin such that the following conditions hold: (1) $V(x)$ is positive definite; (2) There exist real numbers $c > 0$ and $0 < \gamma < 1$ such that $\dot{V}(x) + cV^{\gamma} \leq 0$. Then the origin is locally finite-time stable, and the settling time satisfies*

$$t(x_0) \leq \frac{V(x_0)^{1-\gamma}}{c(1-\gamma)}. \tag{3.39}$$

Chapter 4

Synchronization Control Design for Leader-Following Manipulators under Network-Induced Constraints

This chapter investigates the motion synchronization of a networked leader-following manipulator system. Due to the network-induced constraints as described in Section 3.4.3, we aim at addressing two consequent issues, namely,

- Large phase shift
- Adverse neighbor influence.

Concepts from the NTSM control method (as in Section 3.5) are developed into novel control approaches that not only guarantee semi-global asymptotic stability but also improve the tracking performance in terms of accuracy. Numerical simulations and physical experiments are performed to verify the effectiveness of the proposed approaches[1].

4.1 Problem Description

Let the leader-following manipulator system be formulated by E-L equations of motion of the following forms:

$$M_0(\boldsymbol{q}_0)\ddot{\boldsymbol{x}}_0 + C_0(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0)\dot{\boldsymbol{x}}_0 + \boldsymbol{g}_0(\boldsymbol{q}_0) = \boldsymbol{u}_0, \tag{4.1}$$

$$M_i(\boldsymbol{q}_i)\ddot{\boldsymbol{x}}_i + C_i(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{x}}_i + \boldsymbol{g}_i(\boldsymbol{q}_i) = \boldsymbol{u}_i, \tag{4.2}$$

where the leader and the followers are indexed by the subscripts 0 and i, respectively. $i \geq 2$.

---

[1] This chapter was published as the work in [93], 2168-2216 ©2020 IEEE. Reprinted, with permission, from Henghua Shen, Ya-Jun Pan, Usman Ahmad and Bingwei He, "Pose Synchronization of Multiple Networked Manipulators using Non-singular Terminal Sliding Mode Control", IEEE Transactions on Systems, Man and Cybernetics: Systems, February/2020, and as the work titled "Online Noise-Estimation-based Neighbor Selection for Multi-Manipulator Systems" by H. Shen et al. in Proceedings of the 21st IFAC World Congress, 2020, accepted.

In network systems, communication channels inevitably experience a certain amount of random delays and packet loss. As a result, provided a time-varying leader trajectory ($\boldsymbol{x}_0(t)$) in a time-delayed leader-following system, the followers tend to track the delayed leader signals, that is,

$$\boldsymbol{x}_i(t) \to \boldsymbol{x}_0(t - T_{i0}(t)).$$

Assume that $\boldsymbol{x}_0(t)$ is a periodic signal, the network delays will cause the difference $\Delta_x = \boldsymbol{x}_i(t) - \boldsymbol{x}_0(t - T_{i0}(t))$, which is termed *phase shift* (or *horizontal shift*). When the network is weakly connected and the data transmission is directed [22, 23], the horizontal shift becomes more evident as the followers that have no direct connection with the leader must rely on the delayed states of their connected neighbours and do not react to the change of the leader until its information has been passed through multiple delays.

Another issue being discussed in this chapter is the adverse influence from neighbours. For each agent, it may connect to multiple neighbours and tend to follow the delayed follower states, that is,

$$\boldsymbol{x}_i(t) \to \boldsymbol{x}_j(t - T_{ij}(t)),$$

where $i \neq j$ and $j = 1, 2, \cdots, n$.

In addition to the phase shift effect between the neighbour communication, once the agent i is affected by its neighbour's incorrect information, it, in return, will broadcast the poor updated state information to the neighbours. Consequently, state errors may keep propagating and exacerbating through the MASs.

Without loss of generality, the discussions of these two problems, large phase shift and adverse neighbour influence, obey the following assumptions to facilitate the respective discussions.

Assumption 4.1. *In analysis of the adverse neighbor influence, it only considers the translational position regulation of a group of manipulators' end effectors. Whereas for discussions of the large phase shift problem, the full pose regulation of a group of manipulators' end effectors is considered.*

Assumption 4.2. *In discussions of both issues, dynamics of the manipulators are assumed to be known in the control design.*

Assumption 4.3. *The joint position is measurable by built-in encoders, while the differential states are estimated or obtained via the time differentiation of the position signals when required.*

Assumption 4.4. *The leader's end effector pose is time-varying and known to only a subset of the follower manipulators. In addition, when the pose tracking is realized in the steady state, the differential states (e.g., velocity and acceleration) of the followers are upper-bounded by the maximum assigned velocity and acceleration of the leader [94], that is,* $\|\dot{\boldsymbol{x}}_i(t)\| \leq max(\|\dot{\boldsymbol{x}}_0(t)\|)$ *and* $\|\ddot{\boldsymbol{x}}_i(t)\| \leq max(\|\ddot{\boldsymbol{x}}_0(t)\|)$*, s.t.* $\ddot{\boldsymbol{x}}_i \rightarrow \ddot{\boldsymbol{x}}_0$*.*

Assumption 4.5. *For discussions of the large phase shift problem, all network channels experience a constant and uniform delay, which can be experimentally realized by using a receiver buffer whose size is not less than the maximum network delay to store the incoming data from the neighbours [95, 96]. In analysis of the adverse neighbour influence, communication amongst the followers is strongly connected, while the leader's information is known to only a subset of the follower manipulators. In the data exchange, the transmission is subject to varying delays and random packet loss.*

Control Objective

In discussion of the large phase shift and the adverse neighbour influence, a leader-following nonlinear manipulator system is considered. The general goal is to drive the motion synchronization of multiple followers' end effectors towards a virtual leader's trajectory in a finite time. In addition, nonlinear control schemes are developed in an effort to minimize tracking error between any two agents whose communications are subject to the network-induced constraints. In other words, by using $\boldsymbol{x}$ to generally denote the (translational and/or angular) motion of the end effector, the network-wide objective of this chapter can be described as $\|\boldsymbol{x}_i(t)-\boldsymbol{x}_0(t)\|_1 \leq \Delta_{\epsilon,i0}$ and $\|\boldsymbol{x}_i(t)-\boldsymbol{x}_j(t)\|_1 \leq \Delta_{\epsilon,ij}$. The bounds $\Delta_{\epsilon,i0} \geq 0$ and $\Delta_{\epsilon,ij} \geq 0$ are desired to be as small as possible.

4.2 NTSM Control Method Using Mixed-Type Feedback

In this section, we aim at dealing with the issue of large phase shift caused by the network delay and its equivalences. The novel mixed-type feedback is used in an

NTSM control method to reduce the horizontal shift effect in an effort to improve the pose tracking synchronization between the manipulators' end effectors. The stability analysis is provided to prove the finite-time boundedness of the tracking error signals. In addition, quantitative evaluations of the multiple effects on the error bound are carried out to give insight into the subsequent control parameter selection. Numerical simulation and experimental results are provided to demonstrate the effectiveness of the developed controller.

4.2.1 Types of Feedback Signals

For networked multi-agent systems, three types of feedback signals are commonly used in the control design [52].

Feedback without self-delay

$$\boldsymbol{e}_i(t) = \sum_{j=1}^{n} a_{ij}[\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t - T_{ij})]. \tag{4.3}$$

Feedback with identical (or exact) self-delay

$$\boldsymbol{e}_i(t) = \sum_{j=1}^{n} a_{ij}[\boldsymbol{x}_i(t - T_{ij}) - \boldsymbol{x}_j(t - T_{ij})]. \tag{4.4}$$

Feedback with different (or inexact) self-delay

$$\boldsymbol{e}_i(t) = \sum_{j=1}^{n} a_{ij}[\boldsymbol{x}_i(t - \hat{T}_{ij}) - \boldsymbol{x}_j(t - T_{ij})], \tag{4.5}$$

where $\hat{T}_{ij}$ is the inexact estimation of T_{ij}. a_{ij} is the adjacency matrix element as in (3.30).

The comparative advantages and disadvantages are summarized as follows to understand the trade-off in deciding which type of feedback to be used. In the control design, using the feedback signals without self-delay as in (4.3) is superior in maintaining stability while the tracking performance degrades. In contrast, controllers designed by exploiting the feedback signals with exact self-delay as in (4.4) or inexact self-delay as in (4.5) help improve the tracking performance, but due to the constant use of the delayed controllers, the systems are prone to be unstable, especially when experiencing considerably long network delays. This conflict can be properly managed to reduce, if not eliminate, the phase shift effect. The following remark describes our novel development of the mixed-type feedback signals.

Remark 4.1. *In many applications, we are concerned only with the synchronization of position, while the velocity, which is also known as an important power signal, is managed to guarantee the system stability. Therefore, consider the different effects of the feedback signal as in (4.3)-(4.5), we develop the mixed-type feedback signals by using the self-delayed position feedback and the velocity feedback without self-delay, in attempt to reduce the large phase shift induced by network delays.*

The non-trivial feasibility and effectiveness of using the proposed mixed-type feedback signals will be illustrated in Fig.4.2.

4.2.2 Controller Development

Let $\boldsymbol{x} = [\boldsymbol{p}^{\mathrm{T}}\ \boldsymbol{\xi}^{\mathrm{T}}]^T \in \mathbb{R}^7$ define the full pose of a end effector, where $\boldsymbol{p} \in \mathbb{R}^3$ is the translational position and the unit quaternion $\boldsymbol{\xi} \in \mathbb{R}^4$ represents the orientation. The constant delay is denoted by T. The translational and angular mixed-type feedback signals are developed respectively as follows for the subsequent full-pose NTSM control design.

Translational Mixed-Type Feedback

The translational position error $\boldsymbol{e}_i^{lp} \in \mathbb{R}^3$ and velocity error $\boldsymbol{e}_i^{lv} \in \mathbb{R}^3$ of the i^{th} manipulator are given as

$$\boldsymbol{e}_i^{lp} = \sum_{j=1}^{n} a_{ij}[\boldsymbol{p}_i(t-T) - \boldsymbol{p}_j(t-T)] + b_i[\boldsymbol{p}_i(t-T) - \boldsymbol{p}_0(t-T)], \tag{4.6}$$

$$\boldsymbol{e}_i^{lv} = \sum_{j=1}^{n} a_{ij}[\dot{\boldsymbol{p}}_i - \dot{\boldsymbol{p}}_j(t-T)] + b_i[\dot{\boldsymbol{p}}_i - \dot{\boldsymbol{p}}_0(t-T)]. \tag{4.7}$$

As a result, the errors $\boldsymbol{e}_i^{lp}$ and $\boldsymbol{e}_i^{lv}$ have the following relationship:

$$\dot{\boldsymbol{e}}_i^{lp} = \boldsymbol{e}_i^{lv} - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\dot{\boldsymbol{e}}_i^{lc}, \tag{4.8}$$

$$\dot{\boldsymbol{e}}_i^{lc} = \dot{\boldsymbol{p}}_i - \dot{\boldsymbol{p}}_i(t-T). \tag{4.9}$$

where $\dot{\boldsymbol{e}}_i^{lc} \in \mathbb{R}^3$ is a self-delayed velocity error, which is achievable since the states at the current and previous time steps are all available.

Angular Mixed-Type Feedback

In design of angular error dynamics, recall from Lemma 3.2 in Chapter 3 and the quaternion error operator in (3.17), the angular position error is given as

$$\boldsymbol{e}_i^{ap} = \sum_{j=1}^{n} a_{ij}\Big(\boldsymbol{e}_{ij}^{\xi} - \mathrm{diag}(\boldsymbol{\xi}_u)\mathrm{sgn}(\boldsymbol{e}_{ij}^{\xi})\Big) + b_i\Big(\boldsymbol{e}_{i0}^{\xi} - \mathrm{diag}(\boldsymbol{\xi}_u)\mathrm{sgn}(\boldsymbol{e}_{i0}^{\xi})\Big), \quad (4.10)$$

where $\boldsymbol{\xi}_u = [1 \quad \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^4$. diag$(*)$ returns a diagonal matrix of an input vector. $\boldsymbol{e}_{ij}^{\xi} \in \mathbb{R}^4$ and $\boldsymbol{e}_{i0}^{\xi} \in \mathbb{R}^4$ are given as follows.

$$\begin{aligned}
\boldsymbol{e}_{ij}^{\xi} &= \begin{bmatrix} e_{ij}^{\eta} \\ \boldsymbol{e}_{ij}^{\epsilon} \end{bmatrix} = \boldsymbol{\xi}_i(t-T) \otimes \boldsymbol{\xi}_j^{*}(t-T) \\
&= \begin{bmatrix} \eta_i(t-T)\eta_j(t-T) + (\boldsymbol{\epsilon}_i)^T(t-T)\boldsymbol{\epsilon}_j(t-T) \\ -\eta_i(t-T)\boldsymbol{\epsilon}_j(t-T) + \eta_j(t-T)\boldsymbol{\epsilon}_i(t-T) - S(\boldsymbol{\epsilon}_i(t-T))\boldsymbol{\epsilon}_j(t-T) \end{bmatrix},
\end{aligned}$$

and

$$\begin{aligned}
\boldsymbol{e}_{i0}^{\xi} &= \begin{bmatrix} e_{i0}^{\eta} \\ \boldsymbol{e}_{i0}^{\epsilon} \end{bmatrix} = \boldsymbol{\xi}_i(t-T) \otimes \boldsymbol{\xi}_0^{*}(t-T) \\
&= \begin{bmatrix} \eta_i(t-T)\eta_0(t-T) + (\boldsymbol{\epsilon}_i)^T(t-T)\boldsymbol{\epsilon}_0(t-T) \\ -\eta_i(t-T)\boldsymbol{\epsilon}_0(t-T) + \eta_0(t-T)\boldsymbol{\epsilon}_i(t-T) - S(\boldsymbol{\epsilon}_i(t-T))\boldsymbol{\epsilon}_0(t-T) \end{bmatrix}.
\end{aligned}$$

The angular velocity error is given by

$$\begin{aligned}
\boldsymbol{e}_i^{av} &= \sum_{j=1}^{n} a_{ij}\dot{\mathbb{e}}_{ij}^{\xi} + b_i\dot{\mathbb{e}}_{i0}^{\xi} = \sum_{j=1}^{n} a_{ij}\begin{bmatrix} \dot{\mathbb{e}}_{ij}^{\eta} \\ \dot{\mathbb{e}}_{ij}^{\epsilon} \end{bmatrix} + b_i\begin{bmatrix} \dot{\mathbb{e}}_{i0}^{\eta} \\ \dot{\mathbb{e}}_{i0}^{\epsilon} \end{bmatrix} \\
&= \sum_{j=1}^{n} a_{ij}\begin{bmatrix} \frac{1}{2}(\mathbb{e}_{ij}^{\epsilon})^T\boldsymbol{e}_{ij}^{w} \\ -\frac{1}{2}\mathbb{e}_{ij}^{\eta}I_3 + S(\mathbb{e}_{ij}^{\epsilon})\boldsymbol{e}_{ij}^{w} - S(\mathbb{e}_{ij}^{\epsilon})\boldsymbol{w}_i \end{bmatrix} \\
&\quad + b_i\begin{bmatrix} \frac{1}{2}(\mathbb{e}_{i0}^{\epsilon})^T\boldsymbol{e}_{i0}^{w} \\ -\frac{1}{2}\mathbb{e}_{i0}^{\eta}I_3 + S(\mathbb{e}_{i0}^{\epsilon})\boldsymbol{e}_{i0}^{w} - S(\mathbb{e}_{i0}^{\epsilon})\boldsymbol{w}_i \end{bmatrix}, \qquad (4.11)
\end{aligned}$$

where $\boldsymbol{e}_{ij}^{w} = \boldsymbol{w}_j(t-T) - \boldsymbol{w}_i$, $\boldsymbol{e}_{i0}^{w} = \boldsymbol{w}_0(t-T) - \boldsymbol{w}_i$. $\boldsymbol{w}_j(t-T), \boldsymbol{w}_i, \boldsymbol{w}_0(t-T) \in \mathbb{R}^3$ can be obtained from (3.19). $\mathbb{e}_{ij}^{\eta}$, $\mathbb{e}_{ij}^{\epsilon}$, $\mathbb{e}_{i0}^{\eta}$, and $\mathbb{e}_{i0}^{\epsilon}$ can be acquired from the following error equations:

$$\mathbb{e}_{ij}^{\xi} = \begin{bmatrix} \mathbb{e}_{ij}^{\eta} \\ \mathbb{e}_{ij}^{\epsilon} \end{bmatrix} = \boldsymbol{\xi}_i \otimes \boldsymbol{\xi}_j^{*}(t-T) = \begin{bmatrix} \eta_i\eta_j(t-T) + (\boldsymbol{\epsilon}_i)^T\boldsymbol{\epsilon}_j(t-T) \\ -\eta_i\boldsymbol{\epsilon}_j(t-T) + \eta_j(t-T)\boldsymbol{\epsilon}_i - S(\boldsymbol{\epsilon}_i)\boldsymbol{\epsilon}_j(t-T) \end{bmatrix},$$

$$\mathbb{e}_{i0}^{\xi} = \begin{bmatrix} \mathbb{e}_{i0}^{\eta} \\ \mathbb{e}_{i0}^{\epsilon} \end{bmatrix} = \boldsymbol{\xi}_i \otimes \boldsymbol{\xi}_0^*(t-T) = \begin{bmatrix} \eta_i\eta_0(t-T) + (\boldsymbol{\epsilon}_i)^T\boldsymbol{\epsilon}_0(t-T) \\ -\eta_i\boldsymbol{\epsilon}_0(t-T) + \eta_0(t-T)\boldsymbol{\epsilon}_i - S(\boldsymbol{\epsilon}_i)\boldsymbol{\epsilon}_0(t-T) \end{bmatrix}.$$

Similar to the design of translational errors as in (4.8), a self-delayed angular velocity errors $\boldsymbol{e}_i^{ac} \in \mathbb{R}^4$ is introduced to establish the relationship between the orientation and the angular velocity as

$$\begin{aligned} \boldsymbol{e}_i^{ac} &= \begin{bmatrix} e_{\eta i}^{ac} \\ \boldsymbol{e}_{\epsilon i}^{ac} \end{bmatrix} = \Big[\boldsymbol{\xi}_i - \boldsymbol{\xi}_i(t-T)\Big] \otimes \boldsymbol{\xi}_j^*(t-T) \\ &= \begin{bmatrix} \Delta\eta_i\eta_j(t-T) + (\Delta\boldsymbol{\epsilon}_i)^T\boldsymbol{\epsilon}_j(t-T) \\ -\Delta\eta_i\boldsymbol{\epsilon}_j(t-T) + \eta_j(t-T)\Delta\boldsymbol{\epsilon}_i - S(\Delta\boldsymbol{\epsilon}_i)\boldsymbol{\epsilon}_j(t-T) \end{bmatrix}, \end{aligned} \tag{4.12}$$

where $\Delta\eta_i = \eta_i - \eta_i(t-T)$ and $\Delta\boldsymbol{\epsilon}_i = \boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_i(t-T)$.

Since the i^{th} agent could have connections with multiple agents (including the neighbors and the leader), $\boldsymbol{\xi}_j^*(t-T)$ in (4.12) transmitted from the connected agents is not a consistent value. Therefore, $\boldsymbol{\xi}_j^*(t-T)$ is replaced by an estimated orientation signal $\bar{\boldsymbol{\xi}}_j^*(t-T)$ given as

$$\bar{\boldsymbol{\xi}}_j^*(t-T) = \left[\sum_{j=1, j\neq i}^{n} a_{ij} + b_i\right]^{-1} \sum_{j=0}^{n} \boldsymbol{\xi}_j^*(t-T).$$

Then, (4.12) becomes

$$\begin{aligned} \boldsymbol{e}_i^{ac} &= \Big[\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^*(t-T)\Big] \otimes \bar{\boldsymbol{\xi}}_j^*(t-T) \\ &= \begin{bmatrix} \Delta\eta_i\bar{\eta}_j(t-T) + (\Delta\boldsymbol{\epsilon}_i)^T\bar{\boldsymbol{\epsilon}}_j(t-T) \\ -\Delta\eta_i\bar{\boldsymbol{\epsilon}}_j(t-T) + \bar{\eta}_j(t-T)\Delta\boldsymbol{\epsilon}_i - S(\Delta\boldsymbol{\epsilon}_i)\bar{\boldsymbol{\epsilon}}_j(t-T) \end{bmatrix}. \end{aligned} \tag{4.13}$$

Accordingly, the self-delayed angular velocity error becomes

$$\dot{\boldsymbol{e}}_i^{ac} = \begin{bmatrix} \frac{1}{2}(\boldsymbol{e}_{\epsilon i}^{ac})^T\boldsymbol{e}_{wi}^{ac} \\ -\frac{1}{2}e_{\eta i}^{ac}I_3 + S(\boldsymbol{e}_{\epsilon i}^{ac})\boldsymbol{e}_{wi}^{ac} - S(\boldsymbol{e}_{\epsilon i}^{ac})\Delta\boldsymbol{w}_i \end{bmatrix}, \tag{4.14}$$

where $\boldsymbol{e}_{wi}^{ac} = \bar{\boldsymbol{w}}_i(t-T) - \Delta\boldsymbol{w}_i$.

Eventually, we can stack the translational and angular errors to the complete position error vector, velocity error vector, and self-delayed velocity error vector as

$$\boldsymbol{e}_i^p = \begin{bmatrix} \boldsymbol{e}_i^{lp} \\ \boldsymbol{e}_i^{ap} \end{bmatrix} \in \mathbb{R}^7, \quad \boldsymbol{e}_i^v = \begin{bmatrix} \boldsymbol{e}_i^{lv} \\ \boldsymbol{e}_i^{av} \end{bmatrix} \in \mathbb{R}^7, \quad \dot{\boldsymbol{e}}_i^c = \begin{bmatrix} \dot{\boldsymbol{e}}_i^{lc} \\ \dot{\boldsymbol{e}}_i^{ac} \end{bmatrix} \in \mathbb{R}^7. \tag{4.15}$$

Remark 4.2. *According to the translational and angular error dynamics, the stacked self-delayed velocity error relates the stacked pose error and the velocity error as*

$$\dot{\boldsymbol{e}}_i^p = \boldsymbol{e}_i^v - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\dot{\boldsymbol{e}}_i^c. \tag{4.16}$$

Non-Singular Sliding Mode Controller Design

Based on the NTSM control theory generalized in Section 3.5, the sliding surface, $\boldsymbol{s}_i \in \mathbb{R}^7$, is constructed as a function of the position error and the velocity error as:

$$\boldsymbol{s}_i \;=\; \boldsymbol{e}_i^p + \beta(\boldsymbol{e}_i^v)^{\alpha}, \tag{4.17}$$

where the gains α and β have been defined in (3.36). Then, the task-space controller for the follower manipulators is designed as

$$\begin{aligned}\boldsymbol{u}_i \;=\; & C_i\dot{\boldsymbol{x}}_i + \boldsymbol{g}_i + \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{-1} M_i\Bigg\{ - \frac{(\boldsymbol{e}_i^v)^{(2-\alpha)}}{\alpha\beta} + \frac{1}{\alpha\beta}\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big) \\ & \mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(1-\alpha)}\Big)\dot{\boldsymbol{e}}_i^c + \sum_{j=1}^{n} a_{ij}\ddot{\boldsymbol{x}}_j(t-T) + b_i\ddot{\boldsymbol{x}}_0(t-T) - \kappa_1\mathrm{sgn}(\boldsymbol{s}_i)\Bigg\},\end{aligned} \tag{4.18}$$

where $\boldsymbol{u}_i \in \mathbb{R}^7$, and κ_1 is a positive constant. Because $1-\alpha < 0$, to avoid the mathematical singularity, $(\boldsymbol{e}_i^v)^{(\alpha-1)}$ is operated element-wisely as

$$(e_i^v(m))^{(1-\alpha)} = \begin{cases} (e_i^v(m))^{(1-\alpha)}, & \text{if } e_{i,m}^v \neq 0, \;\; m = 1, 2, \cdots, 7 \\ 0, & \text{otherwise.} \end{cases}$$

The following theorems and remarks describe the tracking performance of the networked multi-manipulator systems through the use of the Lyapunov method.

Theorem 4.1. *For the follower manipulators described in operational space given by (4.2), it is guaranteed to reach the sliding surface in (4.17) and an auxiliary switching surface given by*

$$\|\boldsymbol{e}_i^v\| = \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\|\dot{\boldsymbol{e}}_i^c\|, \;\; sgn(\boldsymbol{e}_i^v) = sgn(\dot{\boldsymbol{e}}_i^c), \;\; s.t.\; \boldsymbol{s}_i = \boldsymbol{0}. \tag{4.19}$$

in a finite time using the controller in (4.18).

See Appendix A.1 for the proof.

Theorem 4.1 implies that bounds of the pose error and velocity error are ultimately related to the self-delayed velocity error. Recall Assumption 4.4, the following proposition and remarks state multiple effects on the upper tracking error bounds.

Proposition 4.1. *In the pose tracking synchronization control, when the sliding surface and the switching surface are all reached, the pose error and the velocity error are bounded as*

$$\|\boldsymbol{e}_i^{lp}\| \leq \beta_i\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{\alpha_i} T^{\frac{\alpha_i}{2}} (A^{lM})^{\alpha_i}, \tag{4.20}$$

$$\|\boldsymbol{e}_i^{ap}\| \leq \beta_i\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{\alpha_i} T^{\frac{\alpha_i}{2}} (A^{aM} + V^{aM})^{\alpha_i}, \tag{4.21}$$

where the translational and angular maximum assigned velocities and accelerations are $\|\dot{\boldsymbol{p}}_i\| \leq V^{lM}$, $\|\dot{\xi}_i\| \leq V^{aM}$, $\|\ddot{\boldsymbol{p}}_i\| \leq A^{lM}$, *and* $\|\ddot{\xi}_i\| \leq A^{aM}$, *and* $V^M = \sqrt{(V^{lM})^2 + (V^{aM})^2}$ *and* $A^M = \sqrt{(A^{lM})^2 + (A^{aM})^2}$.

See Appendix A.2 for the proof.

Remark 4.3. *In the leader-following system where the reference signal is given with* $A_0^M = 0$ *(i.e., the leader is static or moves at a constant speed), the pose tracking error is independent of the effect of network delays, and zero error can be achieved. On the other hand, when the reference signal is given with a non-zero maximum assigned acceleration or a time-varying speed* ($A^M \neq 0$)*, the larger communication delays result in the increased pose error bound.*

Remark 4.4. *In a leader-following network-delayed system, the leader trajectory with a higher maximum assigned acceleration* A^M *increases the pose error bound.*

Remark 4.5. *Selecting higher control gains* β *or smaller value of* α *leads to larger pose error bounds.*

Remark 4.6. *The larger network delay causes larger upper bounds of the pose error.*

4.2.3 Simulation Results

This section provides numerical simulation results of three follower manipulators synchronizing to a leader's state trajectory. The leader and followers are all modeled as

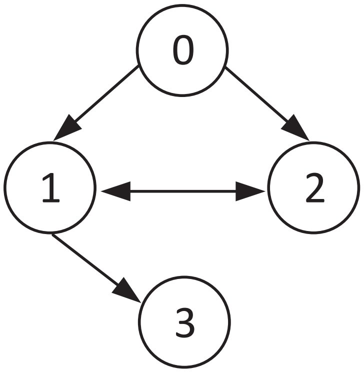


Figure 4.1: Weakly connected topology.

Table 4.1: The maximum assigned velocity and acceleration generated by different angular frequencies f.

f	Translational		Angular	
	V^{lM} (m/s)	A^{lM} (m/s^2)	V^{aM}	A^{aM}
$\pi/40$	0.0867	0.0091	0.0604	0.0050
$\pi/35$	0.0990	0.0119	0.0690	0.0065
$\pi/30$	0.1155	0.0162	0.0805	0.0088
$\pi/25$	0.1384	0.0234	0.0965	0.0127
$\pi/20$	0.1730	0.0365	0.1207	0.0199
$\pi/15$	0.2307	0.0649	0.1606	0.0350

two-link robot manipulators as in Section 3.3.3 and the network topology is shown in Fig.4.1. The joint states of follower manipulators are initialized at different values (unit: rad): $\boldsymbol{q}_1(0) = [1.047 \;\; 1.047]^{\mathrm{T}}$, $\boldsymbol{q}_2(0) = [1.047 \;\; 1.390]^{\mathrm{T}}$, $\boldsymbol{q}_3(0) = [1.500 \;\; 0.500]^{\mathrm{T}}$. All initial velocities are set as zero.

A time-varying reference trajectory in task space is produced from a sinusoidal joint trajectory of $\boldsymbol{q}_0 = [0.8 \;\; 0.8\sin(ft)]^{\mathrm{T}}$ rad through the forward kinematics as in Section 3.2.5. Consider the angular frequency f ranging from $\pi/40$ to $\pi/15$, the resultant translational and angular assigned velocities and accelerations are shown in Table 4.1. The unit of the orientation is omitted as the four quaternion parameters do not have an intuitive physical meaning. In the remaining discussions, the according angular frequency f is used when referring to the assigned velocities and accelerations. In this simulation, the follower controllers use the same control gains, that is, $\alpha_i = \alpha$, $\beta_i = \beta$, and $\kappa_i = \kappa$. Notice that the manipulator models in the simulation are planar,

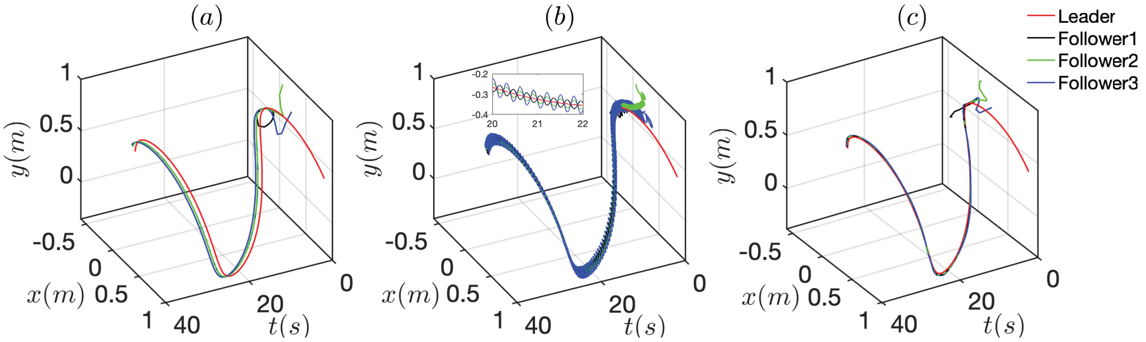


Figure 4.2: Comparison of the translational tracking performance using different types of feedback signals of (a) **Case I**, (b) **Case II**, and (c) **Case III**.

so the z coordinate, $\boldsymbol{\epsilon}_1$, and $\boldsymbol{\epsilon}_2$ are zero.

Comparisons of Using Different Types of Feedback Signals

To present the merits of using the mixed-type feedback in the control design, comparisons with the use of the following two different types of feedback are performed (Taking translational feedback as an example).

Case I: Feedback signals without self-delay

$$\boldsymbol{e}_i^{lp} = \sum_{j=1}^{n} a_{ij}[\boldsymbol{p}_i(t) - \boldsymbol{p}_j(t-T)] + b_i[\boldsymbol{p}_i(t) - \boldsymbol{p}_0(t-T)],$$

$$\boldsymbol{e}_i^{lv} = \sum_{j=1}^{n} a_{ij}[\dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{p}}_j(t-T)] + b_i[\dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{p}}_0(t-T)].$$

Case II: Feedback signals with self-delay

$$\boldsymbol{e}_i^{lp} = \sum_{j=1}^{n} a_{ij}[\boldsymbol{p}_i(t-T) - \boldsymbol{p}_j(t-T)] + b_i[\boldsymbol{p}_i(t-T) - \boldsymbol{p}_0(t-T)],$$

$$\boldsymbol{e}_i^{lv} = \sum_{j=1}^{n} a_{ij}[\dot{\boldsymbol{p}}_i(t-T) - \dot{\boldsymbol{p}}_j(t-T)] + b_i[\dot{\boldsymbol{p}}_i(t-T) - \dot{\boldsymbol{p}}_0(t-T)].$$

Case III: The mixed-type feedback as in (4.6) and (4.7).

Fig.4.2 presents the comparison results of three cases simulated with $f = \pi/15$. As to the network delay, **Case I** and **Case III** are simulated with $T = 0.5\ s$, while the delay in **Case II** is $T = 0.1\ s$ because the delay of $T = 0.5\ s$ results in the tracking failure due to the poor stability.

As can be seen from Fig.4.2(a), merely using the feedback without self-delay (**Case I**) can guarantee the tracking stability while the tracking accuracy degrades with a phase shift caused by time delays. In contrast, in Fig.4.2(b) where the controller utilizes the feedback signals with self-delay (**Case II**), the undesired phase shift is reduced. However, the trajectories of the followers present evident oscillations along the leader trajectory, and such oscillations will lead to instability when the delay increases. Fig.4.2(c) shows benefits of the controller using the mixed-type feedback signals (**Case III**) to improve the tracking accuracy.

Multiple Effects on Tracking Performance

The following results and discussions demonstrate the influences of time delays, angular frequencies, and the selection of control gains. In each discussion, six datasets of the pose error bound are collected and each dataset contains 4000 data. To present the results, the boxplot is used to display the distribution of the pose error. The median of a boxplot is used to denote the average pose error and the interquartile range (IQR) that shows the 25th to the 75th percentile of the distribution is used to demonstrate the error bound. A boxplot also displays the outlier number that demonstrates the suddenly large error w.r.t the median value. Table 4.2$\sim$Table 4.5 present the average values, IQR, and outliers of the pose tracking error. In the following discussions, unless other specified, the control gains are selected as $\alpha = 7/5$, $\beta = 1$, and $\kappa_1 = 6$, the angular frequency is selected as $f = \pi/40$, and the delay is set as $T = 0.05\ s$.

Effect of Network Delays: In discussion of the effect of the network delay, the latency is selected in $0.05\ s \sim 0.3\ s$ according to the physical tests in [88, 89]. As it can be seen from Fig.4.3(a), the tracking errors increase exponentially as the network delay increases from $0.05\ s$ to $0.3\ s$. This verifies our theoretical results in Proposition 4.1, showing that the growth of the pose tracking error is a power function of the increasing time delays. As claimed in Remark 4.3, when the reference signal is given with a time-varying speed, i.e., $A^M \neq 0$, the longer network delay results in the increased pose error bounds. This trend can be seen in Table 4.2. When the delay is small, e.g., $T = 0.05\ s$, the average pose error is very small, that is, $5.902 \times 10^{-5}\ m$, and the pose error is ranging within a very small bound. For example,

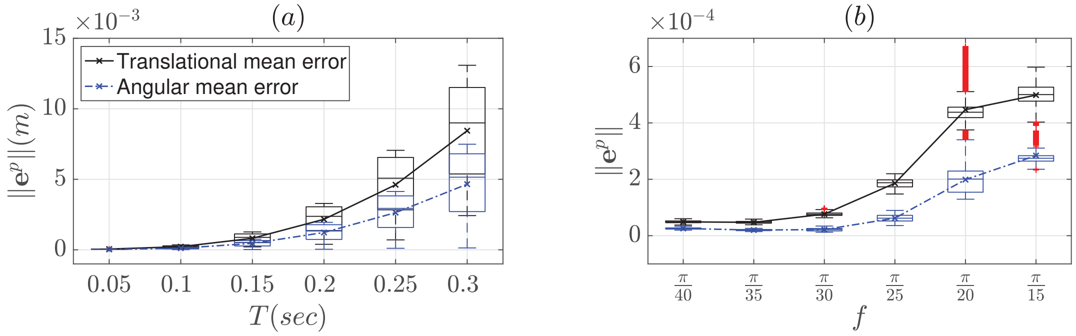


Figure 4.3: The Box plot shows the medians, quartiles, extremes, outliers, and the average of the pose tracking error varying with (a) different delays; (b) different angular frequencies f.

Table 4.2: Results of tracking errors affected by network delays

T (s)	Mean (m)	IQR (m)	Outlier Num.
0.05	5.902×10^{-5}	$[5.454,\ 6.317] \times 10^{-5}$	0
0.10	2.502×10^{-4}	$[1.703,\ 3.225] \times 10^{-4}$	0
0.15	9.362×10^{-4}	$[5.880,\ 13.041] \times 10^{-4}$	0
0.20	2.491×10^{-3}	$[1.532,\ 3.534] \times 10^{-3}$	0
0.25	5.318×10^{-3}	$[3.252,\ 7.586] \times 10^{-3}$	0
0.30	9.676×10^{-3}	$[6.044,\ 13.327] \times 10^{-3}$	0

Table 4.3: Results of tracking errors affected by angular frequencies

f	Mean (m)	IQR (m)	Outlier Num.
$\pi/40$	5.540×10^{-5}	$[5.206,\ 5.934] \times 10^{-5}$	0
$\pi/35$	5.185×10^{-5}	$[4.845,\ 5.534] \times 10^{-5}$	0
$\pi/30$	8.020×10^{-5}	$[7.559,\ 8.402] \times 10^{-5}$	85
$\pi/25$	1.964×10^{-4}	$[1.815,\ 2.102] \times 10^{-4}$	0
$\pi/20$	4.908×10^{-4}	$[4.540,\ 5.106] \times 10^{-4}$	155
$\pi/15$	5.742×10^{-4}	$[5.465,\ 5.961] \times 10^{-4}$	129

the interquartile range is only $[5.454,\ 6.317] \times 10^{-5}$ m. However, when T increases to 0.3 s, the average pose error increases to 9.676×10^{-3} m, which is approximately 200 times higher than that of $T = 0.05$ s. The error bound also significantly increases to $[9.676,\ 13.327] \times 10^{-3}$ m.

Effect of Signal Acceleration: The virtual leader given as the trajectory with $\boldsymbol{q}_0 = [0.8 \quad 0.8\sin(ft)]^{\mathrm{T}}$ rad is with different angular frequency f so as to generate

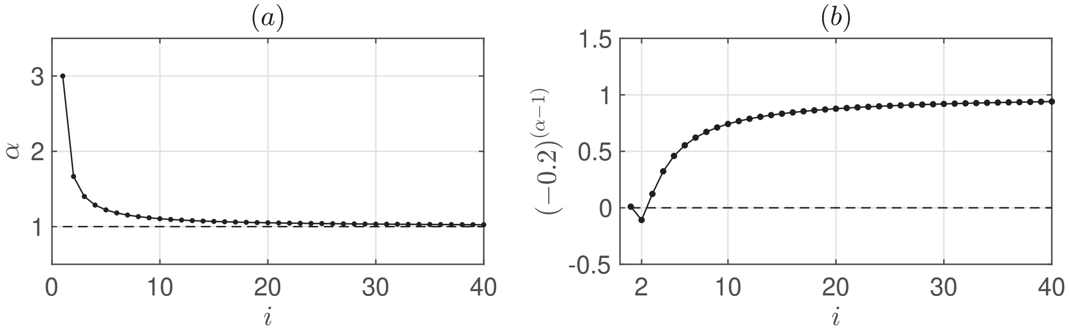


Figure 4.4: Results of (a) α values changing with i and (b) the example calculation, $(-0.2)^{(\alpha-1)}$, with different α values.

different maximum assignment accelerations A^M. The mapping from the angular frequency to the maximum assignment velocity and acceleration is provided in Table 4.1. Fig.4.3(b) demonstrates the distribution of pose errors plotted against the angular frequency f.

As can be seen from Fig.4.3(b), the pose error grows as the angular frequency increases. This trend is expected because the followers become more sluggish to follow the signal with a fast-changing speed when the network is weakly connected and exists time delays. In addition, as the angular frequency increases up to $\pi/20$, it presents obvious outlier (red cross markers), which implies the undesired jitter in the trajectory tracking. Table 4.3 lists the pose errors at different angular frequencies. The behavior in Fig.4.3(b) agrees with the theoretical statements in Proposition 4.1 and Remark 4.4 that the increased A^M enlarges the tracking errors. Therefore, in the physical settings, if the tracking error tends to be higher, it is advisable for the human to operate the task in a slow and smooth manner.

Effect of Control Gains Proposition 4.1 also reveals that the NTSM control gains, α and β, can influence the pose tracking error as well. Notice that another control gain κ is not discussed in this article as its relationship with the tracking error cannot be concluded from (4.20) and (4.21).

Consider the control gain α has a specific form as $\alpha := (2h+1)/(2h-1)$ $(h = 1, 2, \cdots)$, Fig.4.4(a) illustrates the relationship between α and h. α drops significantly over the range of $2 \leq h \leq 20$, but it is approximately equal to 1 when $h > 20$. Moreover, as analyzed in (C.12), α should be selected to guarantee that the rational exponent function $(\boldsymbol{e}_i^v)^{(\alpha_i-1)} \geq \mathbf{0}$ always holds such that the time derivative of the

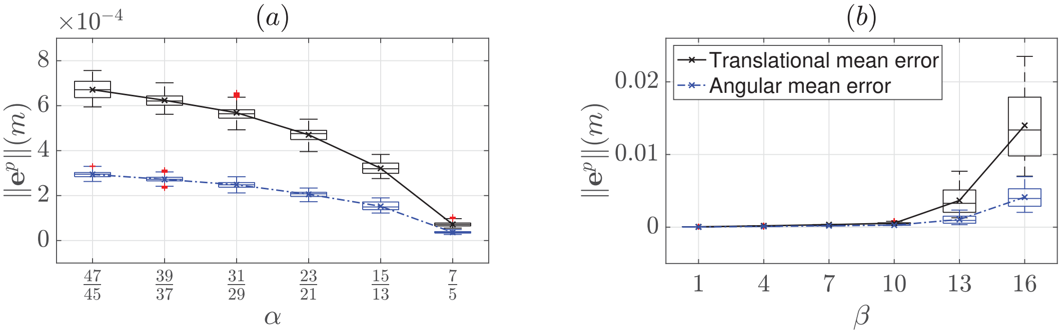


Figure 4.5: The Box plot shows the medians, quartiles, extremes, outliers, and the average of the pose tracking error varying with (a) different α values; (b) different β values.

Table 4.4: Results of tracking errors affected by α.

α	Mean (m)	IQR (m)	Outlier Num.
7/5	8.098×10^{-5}	$[7.317,\ 8.753] \times 10^{-5}$	40
15/13	3.568×10^{-4}	$[3.303,\ 3.854] \times 10^{-4}$	0
23/21	5.179×10^{-4}	$[4.948,\ 5.395] \times 10^{-4}$	0
31/29	6.266×10^{-4}	$[5.998,\ 6.407] \times 10^{-4}$	354
39/37	6.875×10^{-4}	$[6.648,\ 7.084] \times 10^{-4}$	5
47/45	7.400×10^{-4}	$[7.030,\ 7.800] \times 10^{-4}$	0

Table 4.5: Results of tracking errors affected by β.

β	Mean (m)	IQR (m)	Outlier Num.
1	5.747×10^{-5}	$[5.333,\ 6.124] \times 10^{-5}$	5
4	2.162×10^{-4}	$[2.106,\ 2.189] \times 10^{-4}$	261
7	3.946×10^{-4}	$[3.776,\ 4.106] \times 10^{-4}$	0
10	6.169×10^{-4}	$[5.401,\ 6.717] \times 10^{-4}$	109
13	3.891×10^{-3}	$[2.160,\ 5.407] \times 10^{-3}$	0
16	14.740×10^{-3}	$[10.330,\ 18.823] \times 10^{-3}$	0

Lyapunov function is negative semi-definite. Due to the numerical approximation in the Matlab calculation, selecting $\alpha_i = 5/3$ (or $h = 2$) may result in $(\boldsymbol{e}_i^v)^{(\alpha_i - 1)} < \mathbf{0}$ when $\boldsymbol{e}_i^v$ is negative. One illustrative example of this behavior is shown in Fig.4.4(b). Thus, we suggest to select α value when $3 \leq h \leq 20$.

In Fig.4.5(a), the plot of average tracking error presents an exponentially decrease over the increasing α value, while the box width (representing the IQR) does not have significant changes. This verifies that the pose tracking error is an exponential

Table 4.6: The convergence time (unit: s)

β	α			
	7/5	9/7	13/11	25/23
1	2.81	4.25	4.50	5.36
5	7.17	12.42	15.85	20.38
9	10.35	19.61	24.76	31.63
13	12.46	25.87	31.73	41.09

function of α as stated in Proposition 4.1 and Remark 4.5. Therefore, to achieve smaller pose tracking errors, it is preferred to select a greater α value. For example, when $\alpha = 7/5$, the average pose error is only 8.098×10^{-5} m, and the IQR error bound is $[7.317,\ 8.753] \times 10^{-5}$ m, as shown in Table 4.4.

In contrast, for the effect of β, results shown in Fig.4.5 (b) and Table 4.5 demonstrate that a smaller β value allows the smaller pose error. For example, when $\beta = 1$, the error bound is $[5.333,\ 6.124] \times 10^{-5}$ m with the mean error of 5.747×10^{-5} m, while error bound increases to $[10.330,\ 18.823] \times 10^{-3}$ m with the mean error of 14.740×10^{-3} m when $\beta = 16$ is selected. Furthermore, Fig.4.5(b) also shows that selecting β in $[1,\ 10]$ does not seem to have a remarkable effect on the pose tracking error, while the tracking error increases drastically when $\beta > 10$.

Moreover, choosing the control gains, α and β, affects the convergence time t_s. The results are listed in Table 4.6, which shows that the convergence time is shorter when choosing smaller β and greater α, e.g., $\beta = 1$ and $\alpha = 7/5$.

Illustrative Example

Based on the analysis of multiple effects on the tracking accuracy, an illustrative example is simulated with the following control gains: $\alpha = 7/5$, $\beta = 1$, and $\kappa_1 = 4$. The network depicted in Fig.4.1 is utilized with the delay of $T = 0.05$ *sec* and the leader trajectory is given with the angular frequency of $f = \pi/40$.

Taking the simulation results of the follower 1 as an example, as it can be seen in Fig.4.6(a), the auxiliary switching surface (red grid) is reached from around 3.5 *sec*. This practically is the same time when the sliding variable $\boldsymbol{s}$ converges to zero, as shown in Fig.4.9(c). However, after the switching surface is reached, it presents some random jitter during the time interval $[3.5,\ 13]$ *sec* (see the green plot in Fig.4.6(a))

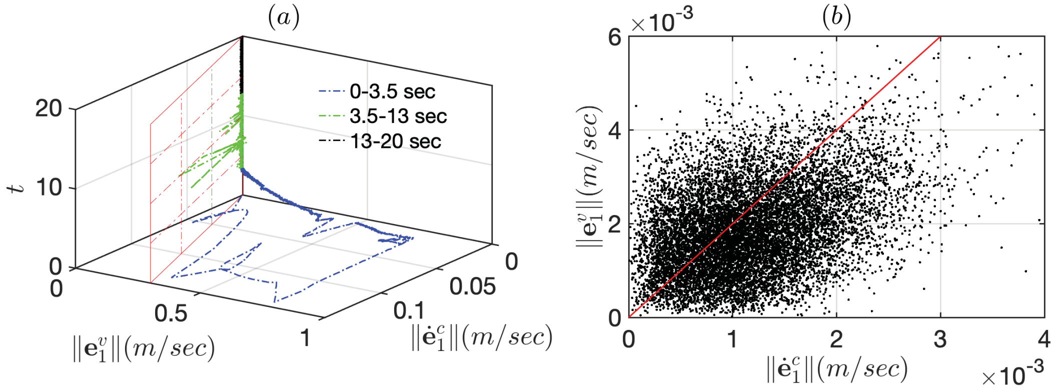


Figure 4.6: (a) The convergence to the auxiliary switching surface (red grid) during the complete time interval $[0,\ 20]$ *sec*. (b) Zoomed plot of the data representing the variable pair $[\|\boldsymbol{e}_1^v\|,\ \|\dot{\boldsymbol{e}}_1^c\|]$ during the time interval $[13,\ 20]$ *sec*, and scattering around the switching surface (red line).

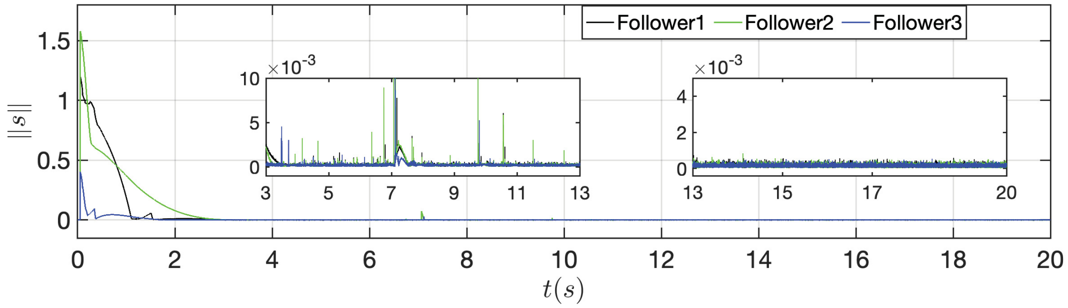


Figure 4.7: The convergence of the sliding surface.

and Fig.4.7 presents the same phenomenon during this time interval. Consequently, a sudden change in the tracking errors can be seen from Fig.4.9 (a) and (b) during the time interval $[3.5,\ 13]$ *sec*.

From $t = 13$ s onwards, the black-color plot Fig.4.6(a) shows that the system settles down and becomes more stable. By zooming in the plot during the time interval $[13,\ 20]$ *sec* in Fig.4.6(a), and casting it into a two-dimensional plane as shown in Fig.4.6(b), a "bee swarm" plot demonstrates the data of the variable pair $[\|\boldsymbol{e}_1^v\|,\ \|\dot{\boldsymbol{e}}_1^c\|]$ scattering around the switching surface as in Theorem 4.1, which is represented by a red line in Fig.4.6(b). This behavior agrees with the results in Fig.4.9 during the time interval $[13,\ 20]$ *sec*, where the tracking error is stable and small bounded, that is, the error bound is $[5.206,\ 5.934] \times 10^{-5}$ m with the mean error of 5.540×10^{-5} m and without any outliers.

The tracking profiles and tracking errors are presented in Fig.4.8 and Fig.4.9,

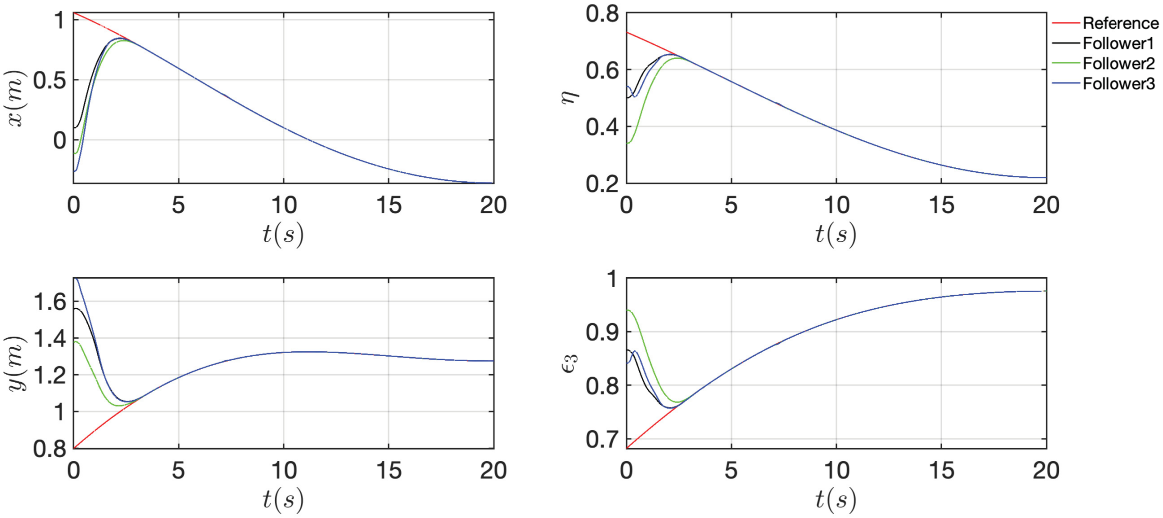


Figure 4.8: The pose tracking synchronization of the illustrative example with $\alpha = 7/5$, $\beta = 1$, $\kappa_1 = 4$, $f = \pi/40$ and $T = 0.05$ *sec*.

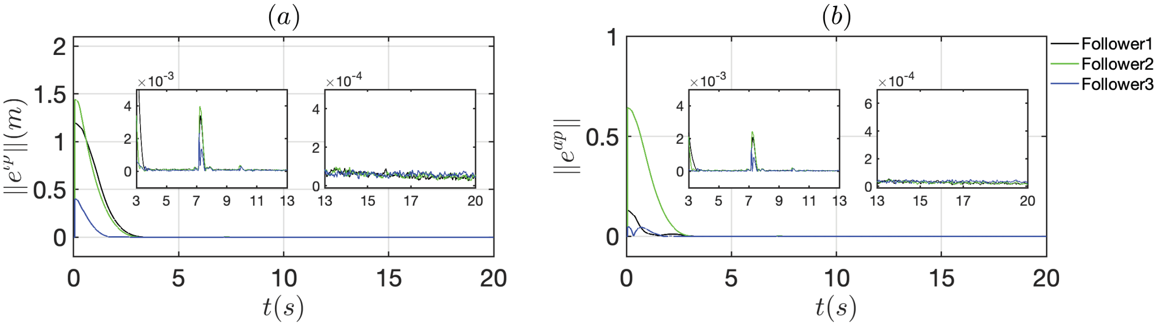


Figure 4.9: (a) The norm of the translational position tracking error; (b) The norm of the angular position tracking error.

respectively. They show that the follower end effectors' poses are able to well synchronize to the leader's in a finite time. Additionally, the controller allows a smooth convergence from their different initial poses.

4.2.4 Experimental Results

To support the practical feasibility of the control theory, experiments were performed using a pair of 3-DOF Phantom Omni haptic devices as two followers and a virtual leader. The topology is given by $A = [0\ 0; 1\ 0]$ and $b = [1,\ 0]$. The experimental setup is shown in Fig.4.10. Two devices are run in two different computers that are connected through an Ethernet LAN. The measured one-way delays of UDP

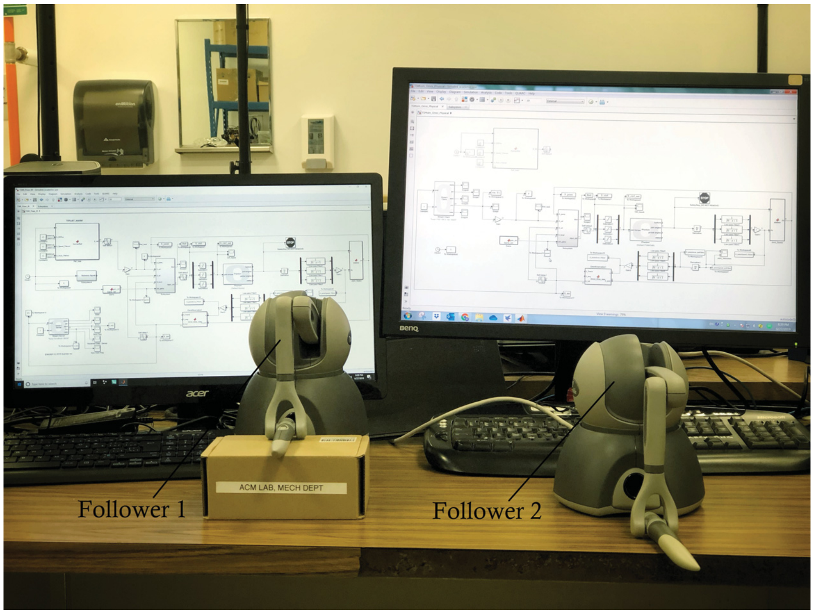


Figure 4.10: The experimental testbed.

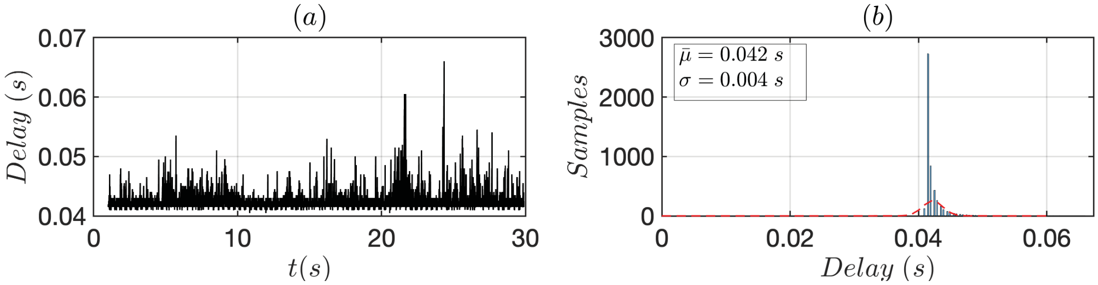


Figure 4.11: (a) Measured one-way network delays in a wired LAN by manually adding a delay of 0.03 s; (b) Mean value of delays and the standard deviation.

send/receive are shown in Fig.4.11 (by manually adding a delay of 0.03 sec), where the average network delay of $\bar{T} = 0.042\ sec$ and the standard error of $\sigma = 0.004\ sec$ are obtained by computing the normal distribution probability density function (ND-pdf). Using the proposed controller with the control gains chosen as $\alpha = 7/5$, $\beta = 1$ and $\kappa = 10$, Fig.4.13 demonstrates the good tracking performance of the systems' translational and angular motions. It can be observed that the translational tracking errors along x, y, and z directions are bounded. The orientation tracking, as well as the orientational tracking error, are depicted in Fig.4.14.

As the control software only provides position measurements, the velocity and acceleration signals are obtained by time-differentiation. As a result, the raw calculations of the velocities and accelerations inevitably suffer from noises and high

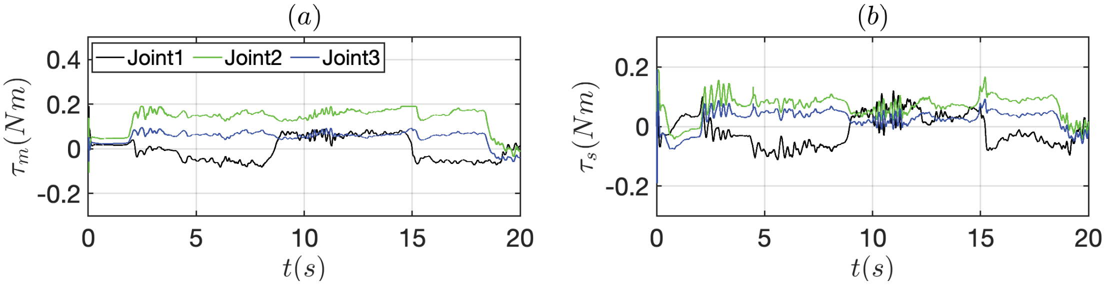


Figure 4.12: Control inputs of the two followers.

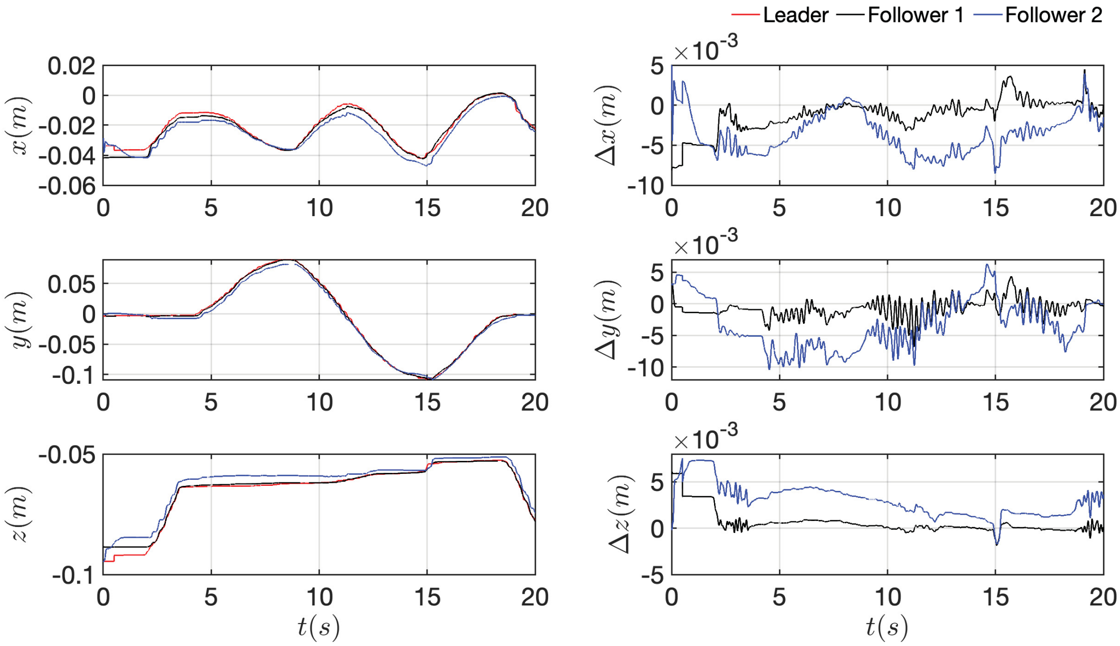


Figure 4.13: The translational position tracking and tracking errors with time-varying reference signals.

frequencies. Therefore, they were filtered by a first order low-pass filter, that is,

$$G(s) = \frac{1}{0.09s + 1}.$$

The filter can greatly reduce the high frequency and chattering effect, as illustrated in Fig.4.15. As depicted in Fig.4.12, the control input signals of the followers are all within the torque limits of the three joints as in Table 3.3. This verifies that our control design in (4.18) is practically feasible. In addition, the sign function in the sliding mode control method is replaced by a saturation function, which helps reduce

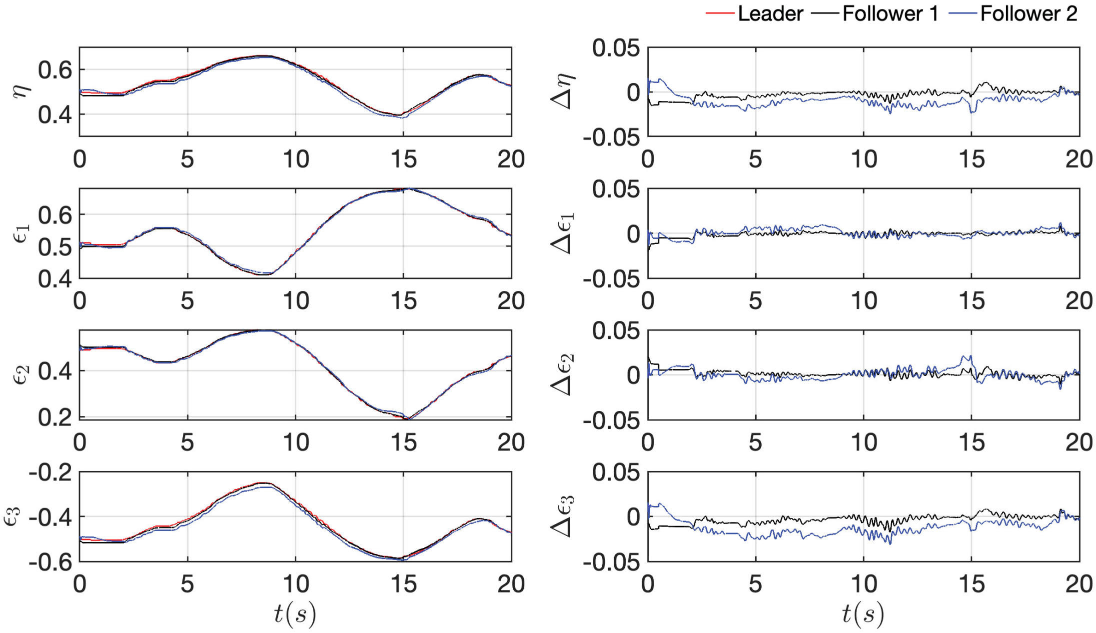


Figure 4.14: The angular position tracking and tracking errors with time-varying reference signals.

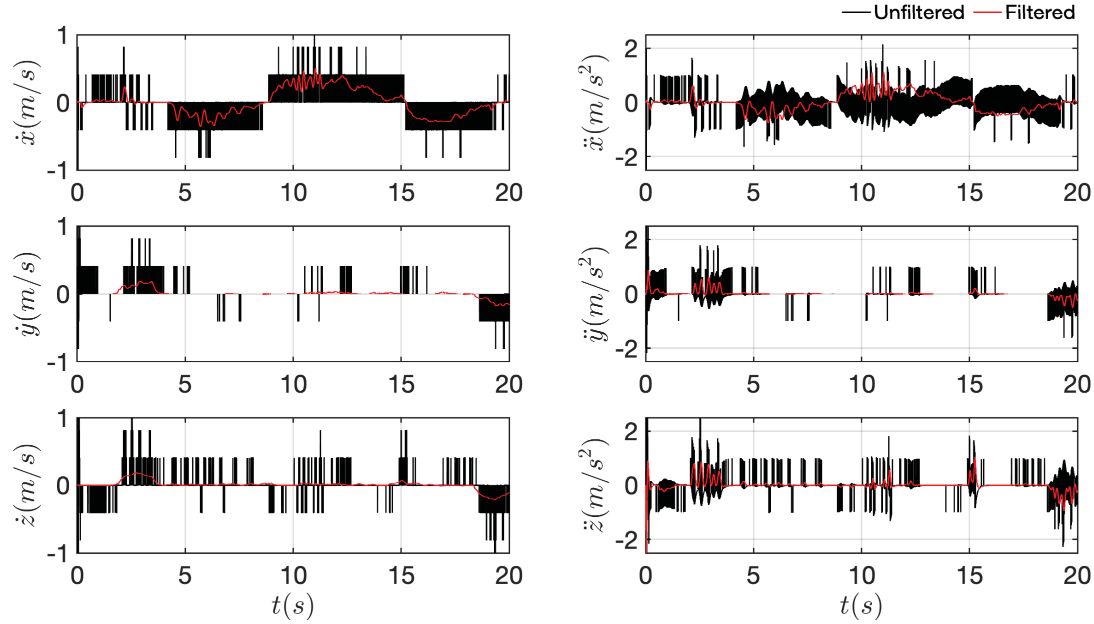


Figure 4.15: The unfiltered (black lines) and filtered (red lines) velocities and accelerations.

the chattering effect in the control signals.

$$\mathrm{sat}(\boldsymbol{s}) = \begin{cases} \|\frac{\boldsymbol{s}}{\Delta}\|_1 \mathrm{sgn}(\boldsymbol{s}), & \|\boldsymbol{s}\|_1 < \Delta, \\ \mathrm{sgn}(\boldsymbol{s}), & \text{otherwise}, \end{cases} \tag{4.22}$$

where Δ is the boundary layer width that provides a proper trade-off between the chattering effect and the tracking accuracy. Here, we select $\Delta = 0.08$.

4.3 Cooperation Control Design with Online Neighbor Selection

In the control design to deal with the adverse neighbour influence, the following two interesting approaches will be introduced into the NTSM controller to enhance each agent's ability to reject the noisy neighbours' influence.

- Continuous-discrete unscented Kalman filter (CD-UKF)
- Energy index-based neighbor-selection policy.

In many engineering and physics applications, a continuous-time signal from a nonlinear system is observed discretely due to processing delays or the manner of operation. Therefore, we consider a continuous-discrete filtering approach [97]. In the nonlinear CD-UKF algorithm, the manipulator dynamics are modelled as continuous-time state-transition functions, while the measurement is regarded as a discrete-time process. The CD-UKF estimates velocity and acceleration readings in joint space, which are then transformed into those in task space via forward kinematics.

To facilitate the description of the CD-UKF, the manipulator dynamics (or process model) are rewritten as:

$$\begin{aligned} \dot{\boldsymbol{X}}_i(t) &= \mathrm{F_m}(\boldsymbol{X}_i(t_k), \boldsymbol{\tau}_i(t_k), t_k) + \boldsymbol{\upsilon}_i(t_{k+1}), \qquad (4.23)\\ \mathrm{F_m}(\boldsymbol{X}_i(t_k), \boldsymbol{\tau}_i(t_k), t_k) &= \begin{bmatrix} \int_{t_k}^{t_{k+1}} \mathrm{F_v}(\boldsymbol{X}_i(\tau), \boldsymbol{\tau}_i(\tau), \tau) d\tau \\ \mathrm{F_v}(\boldsymbol{X}_i(t_k), \boldsymbol{\tau}_i(t_k), t_k) \end{bmatrix}, \\ \mathrm{F_v}(\boldsymbol{X}_i(t_k), \boldsymbol{\tau}_i(t_k), t_k) &= \bar{\mathbb{M}}_i^{-1}(t_k)(\boldsymbol{\tau}_i(t_k) - \bar{\mathbb{C}}_i(t_k)\dot{\boldsymbol{q}}_i(t_k) - \bar{\mathbb{G}}_i(t_k)), \end{aligned}$$

where $\boldsymbol{\tau}_i = J_i^T \boldsymbol{u}_i \in \mathbb{R}^m$ denotes the control inputs in joint space. $\bar{\mathbb{M}}_i$, $\bar{\mathbb{C}}_i$, and $\bar{\mathbb{G}}$ are nominal model parameters. The process noise is denoted by $\boldsymbol{\upsilon}_i(t_k) \in \mathbb{R}^L$. $\mathrm{F_m}$ is a known and noise-free process model. $\boldsymbol{X}_i(t) \in \mathbb{R}^L$ and $\dot{\boldsymbol{X}}_i(t) \in \mathbb{R}^L$ ($L = 2m$) are joint states as

$$\boldsymbol{X}_i(t) = \begin{bmatrix} \boldsymbol{q}_i(t) \\ \dot{\boldsymbol{q}}_i(t) \end{bmatrix}, \quad \dot{\boldsymbol{X}}_i(t) = \begin{bmatrix} \dot{\boldsymbol{q}}_i(t) \\ \ddot{\boldsymbol{q}}_i(t) \end{bmatrix}. \qquad (4.24)$$

The measurement model, $\mathrm{H_m}$, is given by

$$\begin{aligned} \boldsymbol{Y}_{i,k+1} &= \mathrm{H_m}(\boldsymbol{X}_i(t_{k+1}), t_{k+1}) + \boldsymbol{r}_{i,k+1}, \\ \mathrm{H_m}(\boldsymbol{X}_i(t_{k+1}), t_{k+1}) &= D_i \boldsymbol{X}_i(t_{k+1}), \end{aligned} \tag{4.25}$$

where $\boldsymbol{Y}_{i,k+1} \in \mathbb{R}^{\frac{L}{2}}$ is the actual measurement and the measurement noise is denoted by $\boldsymbol{r}_{i,k} \in \mathbb{R}^{\frac{L}{2}}$. D_i is the measurement model coefficient.

In addition to Assumption 4.1 ∼ Assumption 4.5, the following assumptions are made in discussion of the CD-UKF and the neighbor selection policy.

Assumption 4.6. *Assume that the noises in the state transition and observation are additive zero-mean Gaussian processes, that is,* $\boldsymbol{v}_i(t_k) \sim N(0, V_i(t_k))$ *and* $\boldsymbol{r}_{i,k} \sim N(0, R_{i,k})$. $V_i(t_k)$ *and* $R_{i,k}$ *are noise covariance matrices.*

Assumption 4.7. *Assume that only joint positions are measurable such that*

$$D_i = \left[\begin{array}{cc} I_{\frac{L}{2}\times\frac{L}{2}} & 0_{\frac{L}{2}\times\frac{L}{2}} \end{array} \right], \quad D_i \in \mathbb{R}^{\frac{L}{2}\times L}.$$

Assumption 4.8. *[86] The manipulator model is non-redundant and in a non-singular configuration, so that the relationship between the dynamics matrices/vector in joint space (i.e.,* $\bar{\mathbb{M}}_i \in \mathbb{R}^{m\times m}$, $\bar{\mathbb{C}}_i \in \mathbb{R}^{m\times m}$, $\bar{\mathbb{G}}_i \in \mathbb{R}^{m}$*) and the nominal matrices/vector* $\bar{M}_i$,$\bar{C}_i$, *and* $\bar{\boldsymbol{g}}_i$ *as in (3.28) can be described by* $\bar{M}_i = J_i^{-T}\bar{\mathbb{M}}_i J_i^{-1}$, $\bar{C}_i = J_i^{-T}\bar{\mathbb{C}}_i J_i^{-1} - J_i^{-T}\bar{\mathbb{M}}_i J_i^{-1}\dot{J}_i J_i^{-1}$, *and* $\bar{\boldsymbol{g}}_i = J_i^{-T}\bar{\mathbb{G}}_i$.

4.3.1 Continuous-Discrete Unscented Kalman Filter

The CD-UKF algorithm is iteratively implemented with two steps: a prediction step and an update step [97], which will be outlined in this section. The UKF uses the idea of unscented transform that is to deterministically choose a set of sample points, named *sigma points*. The unscented transform will use the following weights.

$$\begin{aligned} W^{\mathrm{m}} &= [w_0^{\mathrm{m}} \ w_1^{\mathrm{m}} \ \cdots \ w_{2L}^{\mathrm{m}}]^{\mathrm{T}}, \\ W^{\mathrm{c}} &= [w_0^{\mathrm{c}} \ w_1^{\mathrm{c}} \ \cdots \ w_{2L}^{\mathrm{c}}]^{\mathrm{T}}, \\ W^{\mathrm{mc}} &= (I - [W^{\mathrm{m}} \ W^{\mathrm{m}} \ \cdots \ W^{\mathrm{m}}])\mathrm{diag}(W^{\mathrm{c}})(I - [W^{\mathrm{m}} \ W^{\mathrm{m}} \ \cdots \ W^{\mathrm{m}}])^{\mathrm{T}}, \end{aligned}$$

where $I \in \mathbb{R}^{(2L+1)\times(2L+1)}$ is an identity matrix. $W^{\mathrm{mc}} \in \mathbb{R}^{(2L+1)\times(2L+1)}$, $W^{\mathrm{m}} \in \mathbb{R}^{2L+1}$ and $W^{\mathrm{c}} \in \mathbb{R}^{2L+1}$.

$$\begin{aligned}
w_0^{\mathrm{m}} &= \frac{\lambda}{L+\lambda}, \\
w_0^{\mathrm{c}} &= \frac{\lambda}{L+\lambda} + (1-\mu^2+\varsigma), \\
w_p^{\mathrm{m}} &= w_p^{\mathrm{c}} = \frac{1}{2(L+\lambda)}, \quad p = 1, 2, \cdots, 2L,
\end{aligned}$$

and $\lambda = \mu^2(L+\varkappa) - L$. Guideline to select the scalars of $\varkappa$, μ, and ς can be found in [98].

Prediction step

In the prediction step, the probability density at time t_{k+1} is predicted from the posterior probability density at time t_k as the boundary condition. Letting $\bar{\boldsymbol{X}}_i(t_k)$ and $P_i(t_k)$ denote the mean and covariance of $\boldsymbol{X}_i(t_k)$, the calculation of sigma points, $\chi_i(t_k) \in \mathbb{R}^{L\times(2L+1)}$, is performed as follows.

$$\chi_i(t_k) = [\bar{\boldsymbol{X}}_i(t_k) \;\; \bar{\boldsymbol{X}}_i(t_k) \;\; \cdots \;\; \bar{\boldsymbol{X}}_i(t_k)] + c\,[0_{L\times L} \;\; \mathrm{chol}(P_i(t_k)) \;\; -\mathrm{chol}(P_i(t_k))],$$

where $c = \sqrt{L+\lambda}$ and chol$(\cdot)$ represents the Cholesky factorization.

The differential equations of the predicted mean and covariance of the state at time instance t_k are written as

$$\begin{aligned}
\dot{\bar{\boldsymbol{X}}}_i^-(t_k) &= \mathrm{F_m}(\chi_i(t_k), \boldsymbol{\upsilon}_i(t_k), \boldsymbol{\tau}_i(t_k))W^{\mathrm{m}}, && (4.26)\\
\dot{P}x_i^-(t_k) &= \chi_i(t_k)W^{\mathrm{mc}}\mathrm{F}_{\mathrm{m}}^T(\chi_i(t_k), \boldsymbol{\tau}_i(t_k)) \\
&\quad + \mathrm{F_m}(\chi_i(t_k), \boldsymbol{\tau}_i(t_k))W^{\mathrm{mc}}\chi_i^T(t_k) + V_i(t_k). && (4.27)
\end{aligned}$$

Then, by integrating the equations in (4.26) and (4.27) over $[t_k, \;\; t_{k+1}]$, we have the predicted mean and covariance as

$$\bar{\boldsymbol{X}}_i^-(t_{k+1}) = \int_{t_k}^{t_{k+1}} \dot{\bar{\boldsymbol{X}}}_i^-(\tau)d\tau,$$

$$Px_i^-(t_{k+1}) = \int_{t_k}^{t_{k+1}} \dot{P}x_i(\tau)d\tau.$$

Update step

In the measurement update step, we continue with computing the sigma points with the predicted mean and covariance from the prediction step, that is,

$$\begin{aligned}\chi_i^-(t_{k+1}) = [&\bar{\boldsymbol{X}}_i^-(t_{k+1}) \;\; \bar{\boldsymbol{X}}_i^-(t_{k+1}) \;\; \cdots \;\; \bar{\boldsymbol{X}}_i^-(t_{k+1})] \\ &+c\Big[0_{L\times L} \;\; \mathrm{chol}(Px_i^-(t_{k+1})) \;\; -\mathrm{chol}(Px_i^-(t_{k+1}))\Big].\end{aligned}$$

Note that measurements are implemented in discrete time, so the mean ($\hat{\boldsymbol{Y}}_{i,k+1}$) and covariance ($\bar{P}y_{i,k+1}$) of the measurements are predicted in a discrete-time manner as well.

$$\begin{aligned}\mathbb{Y}_{i,k+1}^- &= \mathrm{H_m}(\chi_i^-(t_{k+1}), t_{k+1}), \\ \hat{\boldsymbol{Y}}_{i,k+1} &= \mathbb{Y}_{i,k+1}^- W^{\mathrm{m}}.\end{aligned}$$

In the standard UKF process [97], covariance of the measurement is given by

$$\bar{P}y_{i,k+1} = \mathbb{Y}_{i,k+1}^- W^{\mathrm{mc}} (\mathbb{Y}_{i,k+1}^-)^T + R_{i,k+1},$$

and we denote the predicted measurement residual vector as

$$\Delta\boldsymbol{Y}_{i,k+1} = \boldsymbol{Y}_{i,k+1} - \hat{\boldsymbol{Y}}_{i,k+1}.$$

Then predicted covariance of the measurement becomes

$$Py_{i,k+1} = \mathbb{Y}_{i,k+1}^- W^{\mathrm{mc}} (\mathbb{Y}_{i,k+1}^-)^T + R_{i,k+1}. \tag{4.28}$$

The cross-covariance of the state and measurement is

$$Pxy_{i,k+1} = \chi_i^-(t_{k+1}) W^{\mathrm{mc}} (\mathbb{Y}_{i,k+1}^-)^T.$$

Eventually, the mean and covariance of the state at time t_{k+1} are updated by

$$\begin{aligned}K_{k+1} &= Pxy_{i,k+1} Py_{i,k+1}^{-1}, \\ \bar{\boldsymbol{X}}_i^+(t_{k+1}) &= \bar{\boldsymbol{X}}_i^-(t_{k+1}) + K_{k+1}\Delta\boldsymbol{Y}_{i,k+1}, \\ Px_i^+(t_{k+1}) &= Px_i^-(t_{k+1}) - K_{k+1}\bar{P}y_{i,k+1}K_{k+1}^T,\end{aligned}$$

where K_{k+1} denotes the filter gain.

4.3.2 Noise Estimation

Another common assumption that we attempts to relax is that the noise covariances in (4.27) and (4.28) are given as a priori. This does not always hold in practice. Therefore, we introduce the online covariance estimation method [99] into the CD-UKF algorithm, and it is outlined as follows.

Letting $\epsilon_{i,t_k} \in \mathbb{R}^1$ and $\Phi_{i,t_k} \in \mathbb{R}^{L\times L}$ define two prior parameters (shape and rate, respectively) of an inverse Wishart distribution (IWD), their posteriors are computed by

$$\epsilon_{i,t_{k+1}} = \epsilon_{i,t_k} + 1, \tag{4.29}$$

$$\Phi_{i,t_{k+1}} = \Phi_{i,t_k} + \Delta \boldsymbol{S}_i(t_{k+1}) \Delta \boldsymbol{S}_i^T(t_{k+1}), \tag{4.30}$$

where $\Delta \boldsymbol{S}_i(t_{k+1})$ is the Gaussian noise sample given as follows. To estimate V_i in (4.27), we use

$$\Delta \boldsymbol{S}_i(t_{k+1}) = \boldsymbol{X}_i^+(t_{k+1}) - \mathrm{F_m}(\boldsymbol{X}_i^+(t_k), \boldsymbol{\tau}_i(t_k)). \tag{4.31}$$

To estimate R_i in (4.28), we use

$$\Delta \boldsymbol{S}_i(t_{k+1}) = \boldsymbol{Y}_{i,k+1} - \hat{\boldsymbol{Y}}_{i,k+1}. \tag{4.32}$$

Then, along with the CD-UKF procedure, the noise covariance matrices are iteratively approximated by a point estimation as

$$\boldsymbol{\omega}_{i,t_{k+1}} = \frac{\mathrm{Diag}(\Phi_{i,t_{k+1}})}{\epsilon_{i,t_{k+1}} + 1 + p}, \tag{4.33}$$

where p is a positive constant. Diag($*$) returns a vector that contains all diagonal entries of an input square matrix. Notice that $\boldsymbol{\omega}_{i,t_{k+1}}$ represents the estimations of V_i or R_i at time t_{k+1}, subject to the choice of $\Delta \boldsymbol{S}_i(t_{k+1})$ from (4.31) or (4.32).

4.3.3 Neighbor Selection Policy

Although the CD-UKF has been proved to successfully provide smooth estimations, we found that it may not be advantageous to use the estimated states in nonlinear closed-loop control schemes. The estimations may cause fast accumulation of the mechanical energy that leads to temporary or ultimate instability. This effect exacerbates in multi-manipulator systems due to the interplay of agents as well as network-induced constraints (e.g., delays and packet loss). A "performing-well" agent may be affected by its neighbour's incorrect information, and it, in return, will broadcast the poor updated state information to the neighbours. Consequently, state errors can keep propagating and increasing through the MASs. In what follows, we propose an energy-based neighbour selection policy to regulate the interaction between agents in the multi-manipulator system.

Lemma 4.1. *[100] Consider a robot system as a Port-Hamiltonian System connected to its actuators through power ports, the rate of change of its Hamiltonian is the supplied power, that is,* $\boldsymbol{\tau}_i\dot{\boldsymbol{q}}_i$*, where torque* $\boldsymbol{\tau}_i$ *and rotational velocity* $\dot{\boldsymbol{q}}_i$ *are port variables. With the damping dissipation, the energy balance equation can be expressed as*

$$\underbrace{\mathcal{H}_i(t_{p+1}) - \mathcal{H}_i(t_p)}_{\text{Stored Energy}} = \int_{t_p}^{t_{p+1}} [\underbrace{\dot{\boldsymbol{q}}_i^T(t)\boldsymbol{\tau}_i(t)}_{\text{Supplied Energy}} \underbrace{-\dot{\boldsymbol{q}}_i^T(t)\frac{\partial \mathcal{F}_i(\dot{\boldsymbol{q}}_i)}{\partial \dot{\boldsymbol{q}}_i}}_{\text{Dissipated Energy}}]dt,$$

where $\mathcal{F}_i$ *represents the linear friction in joint space [101]. Notice that* t_{p+1} *denotes the current time instant.*

Remark 4.7. *For a dissipative manipulator system that is supplied with excessive energy between two times* t_p *and* t_{p+1}*, the storage energy will exceed the dissipation. Consequently, the manipulation performance would be temporarily perturbed since the excessive kinematic energy or potential energy drives the system away from the equilibrium states.*

Based on Lemma 4.1 and Remark 4.7, we monitor the stored mechanical energy that is estimated by

$$E_i(t_{p+1}) = \int_{t_p}^{t_{p+1}} \hat{\dot{\boldsymbol{q}}}_i(t)\boldsymbol{\tau}_i(t)dt - B\|\hat{\dot{\boldsymbol{q}}}_i(t)\|_2, \tag{4.34}$$

where $E_i(t_{p+1}) \in \mathbb{R}^m$. Let $T_{ns} = t_{p+1} - t_p$ denote the time interval of updating the selection law that will be introduced in Theorem 4.2. T_{ns} is user-defined and not less than sampling time of sensor measurements and assume that t_p is an integral multiple of t_k.

Remark 4.8. *By monitoring the stored energy between two times* t_p *and* t_{p+1}*, as in (4.34), a drastic motion change can be expected if the storage energy appears to* *accumulate or oscillate considerably. Therefore, the temporary storage energy can be used as an indicator of operating performance of the manipulator.*

To facilitate the use of the stored mechanical energy as an indicator, it is normalized by

$$\boldsymbol{\varpi}_i = \frac{\text{sat}(\|E_i(t_{p+1})\|) - \mathbb{E}_{M,i}}{\mathbb{E}_{M,i} - \mathbb{E}_{m,i}}, \quad 0 \leq \boldsymbol{\varpi}_i \leq 1, \tag{4.35}$$

where $\text{sat}(*)$ represents a saturation function, and E_i is then clipped to the upper and lower bounds that are denoted by $\mathbb{E}_M$ and $\mathbb{E}_m$, respectively. At this stage, $\mathbb{E}_M$ and $\mathbb{E}_m$ are chosen at the designer's discretion and the individual robots' application-specific performance requirements. The energy index $\boldsymbol{\varpi}_i$ will be attached to the network packets that are delivered to other connected neighbours.

Remark 4.9. *When $\boldsymbol{\varpi}_i$ is approaching to 0, it indicates the significant mechanical energy accumulation during times t_p and t_{p+1}, while less mechanical energy is stored when $\boldsymbol{\varpi}_i$ is approaching to 1.*

Theorem 4.2. *For the i^{th} manipulator in MASs, when receiving its neighbours' energy indexes, it selects its neighbours based on the neighbour selection indexes given by*

$$\Omega_j = \begin{cases} \frac{1}{2}[1 + sgn(\boldsymbol{\varpi}_{d,i} - a_{ij}\boldsymbol{\varpi}_j)], & if\ max(W_i) \geq \boldsymbol{\varpi}_{d,i} \\ (1 - b_i)[1 - sgn(max(W_i) - a_{ij}\boldsymbol{\varpi}_j)], & otherwise \end{cases}$$

where $\boldsymbol{\varpi}_{d,i}$ is the desired energy index. $W_i = [a_{ij}\boldsymbol{\varpi}_j]$ defines a vector of receiving energy indexes from its neighbours. $min(*)$ and $max(*)$ return minimum and maximum values of the input. $sgn(*)$ represents the sign function.

Remark 4.10. *When $max(W_i) < \boldsymbol{\varpi}_{d,i}$, it implies that there is no eligible neighbour to select. Nevertheless, to avoid an agent to completely disconnect to the MASs, this agent compromises to use the information of either the leader (if $b_i = 1$) or the "least-worst" neighbor (if $b_i = 0$).*

Remark 4.11. *Mostly, 0 and 1 are assigned to Ω_j. For the occasional case where $a_{ij}\boldsymbol{\varpi}_j = \boldsymbol{\varpi}_{d,i}$, we obtain $\Omega_j = 1/2$, which means that "50%" of the neighbor information is used in the controller. One may want to use the full percent of the neighbor information by manually multiplying Ω_j by 2 under the condition of $a_{ij}\boldsymbol{\varpi}_j = \boldsymbol{\varpi}_{d,i}$.*

4.3.4 NTSM Controller Design

With the neighbour selection policy given in Theorem 4.2, the error dynamics are given as follows.

$$\begin{aligned}
\boldsymbol{e}_i &= \sum_{j=1}^{n} \Omega_j(\hat{\boldsymbol{p}}_i(t) - \hat{\boldsymbol{p}}_j(t - T_{ij}(t))) + b_i(\hat{\boldsymbol{p}}_i(t) - \boldsymbol{p}_0(t - T_{i0}(t)))), && (4.36)\\
\dot{\boldsymbol{e}}_i &= \sum_{j=1}^{n} \Omega_j(\hat{\dot{\boldsymbol{p}}}_i(t) - \hat{\dot{\boldsymbol{p}}}_j(t - T_{ij}(t)))) + b_i(\hat{\dot{\boldsymbol{p}}}_i(t) - \dot{\boldsymbol{p}}_0(t - T_{i0}(t))), && (4.37)
\end{aligned}$$

where $\boldsymbol{e}_i \in \mathbb{R}^3$ and $\dot{\boldsymbol{e}}_i \in \mathbb{R}^3$. Through the forward kinematics, $\hat{\boldsymbol{x}}$ and $\hat{\dot{\boldsymbol{p}}}$ are acquired from the joint states estimated by the CD-UKF algorithm. Since $\partial\hat{\boldsymbol{p}}_j(t - T_{ij}(t))/\partial t$ gives the velocity by taking the time-derivative of the estimated and delayed position, while $\hat{\dot{\boldsymbol{p}}}_j(t - T_{ij}(t)))$ is the delayed velocity generated from the CD-UKF method. The following assumption is made to facilitate the stability analysis.

Assumption 4.9. *Assume that* $\hat{\dot{\boldsymbol{p}}}_j(t - T_{ij}(t))) = \partial\hat{\boldsymbol{p}}_j(t - T_{ij}(t))/\partial t$ *such that* $\dot{\boldsymbol{e}}_i = \partial\boldsymbol{e}_i/\partial t$. *Furthermore, it is assumed that* $\hat{\ddot{\boldsymbol{p}}}_j(t - T_{ij}(t))) = \partial\hat{\dot{\boldsymbol{p}}}_j(t - T_{ij}(t))/\partial t$ *such that* $\ddot{\boldsymbol{e}}_i = \partial\dot{\boldsymbol{e}}_i/\partial t$.

Substitute the errors defined in (4.36) and (4.37) into the sliding surface in the form as in (3.36), we have

$$\boldsymbol{s}_i = \boldsymbol{e}_i + \beta_i(\dot{\boldsymbol{e}}_i)^{\alpha_i}. \tag{4.38}$$

Theorem 4.3. *For the leader-following manipulator system described in (4.1) and (4.2), the proposed neighbor selection policy can be incorporated into a NTSM control that is designed as follows:*

$$\begin{aligned}
\boldsymbol{u}_i \;=\;& C_i\hat{\dot{\boldsymbol{p}}}_i + \boldsymbol{g}_i + \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)^{-1} M_i\Big\{\frac{-(\dot{\boldsymbol{e}}_i)^{(2-\alpha_i)}}{\alpha_i\beta_i} + \sum_{j=1}^{n}\Omega_j\hat{\ddot{\boldsymbol{p}}}_j(t - T_{ij}(t))\\
&+ b_i\ddot{\boldsymbol{p}}_0(t - T_{i0}(t)) - \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)\bar{C}_i\mathbb{L}_i M_i^{M+} sgn(\boldsymbol{s}_i) - \kappa_i sgn(\boldsymbol{s}_i)\Big\}, \quad (4.39)
\end{aligned}$$

where i *denotes the* i^{th} *follower.* $\kappa_i > 0$. $\mathbb{L}_i$ *is the upper bound of the velocity estimation error, given by* $\mathbb{L}_i \geq \|\Delta_{\dot{\boldsymbol{p}}}\|_1$ *and* $\Delta_{\dot{\boldsymbol{p}}} = \hat{\dot{\boldsymbol{p}}}_i - \dot{\boldsymbol{p}}_i$. $M_i^{M+} \geq \|M_i^{+}\|_1$ *where* M_i^{+} *represents the Pseudo-inverse of* $\bar{M}_i$.

The stability proof is provided in Appendix A.3.

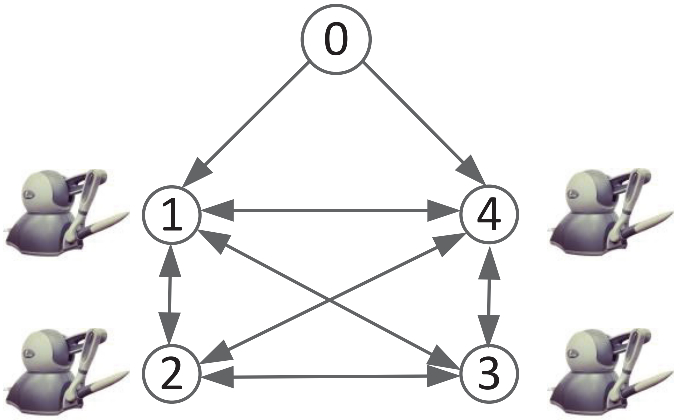


Figure 4.16: Leader-following network topology.

4.3.5 Simulation Results

The numerical simulations were performed with a mathematical model of the 3-DOF Phantom Omni haptic devices described in Section 3.3.3. The leader trajectories were pre-recorded from the Phantom Omni haptic device. The same control gains were chosen for all agents' controllers as $\alpha_i = 7/5$ and $\beta_i = 1$, $\kappa_i = 8$. $T_{ns} = 0.1\,\text{sec}$. $\boldsymbol{\xi} = [0.2\ 0.2\ 0.2]^T$. $\varkappa = 0$, $\mu = 1 \times 10^{-3}$, and $\varsigma = 2$. $p = 1$ as in (4.33). We take one leader, four followers as an example ($n = 4$), and assume that the followers are fully connected, as shown in Fig.4.16. Therefore, we have the adjacency matrices of A and b as

$$A \;=\; \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}.$$

The delays were $0.03 \pm 0.005\ s$ and the packet loss rate was set to 20%. The measurement and process noise variances were selected as: $R = [0.1257\ 0.2000\ 0.3200]$ and $Q = \text{diag}([0.01\ 0.01\ 0.01\ 0\ 0\ 0\ 0\ 0\ 0])$, respectively. As in (4.22), the saturation function is used to reduce the chattering effect.

In what follows, we will present the estimating performance of the CD-UKF algorithm, the adverse effect of agent interplay in a multi-manipulator system, and the benefit of using the proposed neighbour selection policy. Four cases as follows will be discussed and compared:

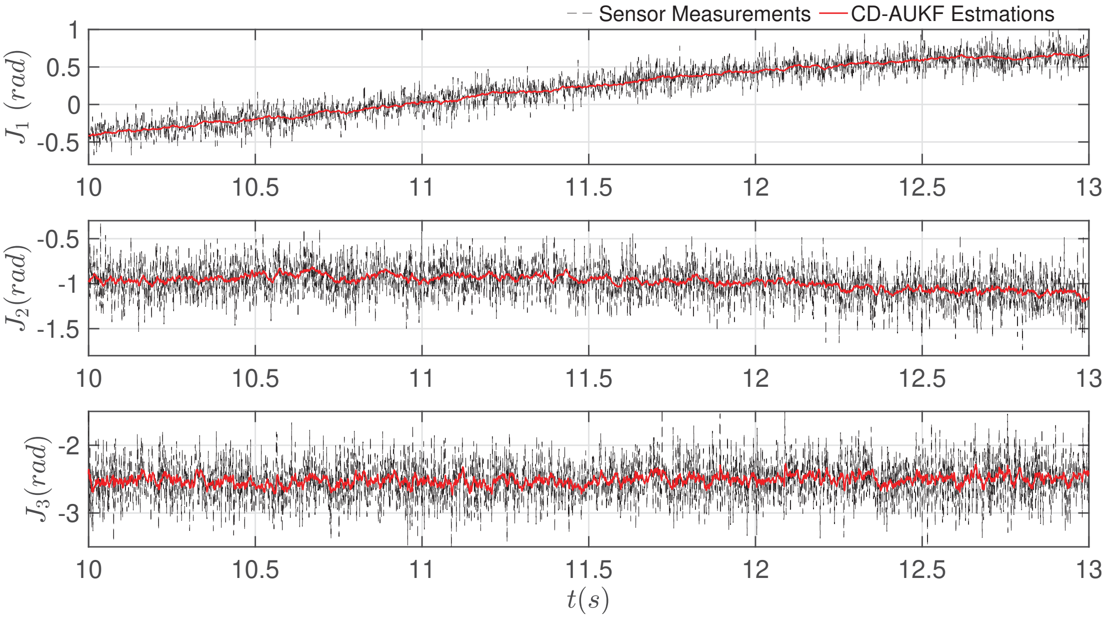


Figure 4.17: The CD-UKF has a good ability to obtain the smooth average state from the noisy measuring signals.

Case 1: The NTSM control for one single manipulator without the CD-UKF algorithm in the closed-loop;

Case 2: The NTSM control for one single manipulator with the CD-UKF algorithm in the closed-loop;

Case 3: The NTSM control for a networked multi-manipulator system as in Fig.4.16 with the CD-UKF algorithm in the closed-loop but without the proposed neighbour selection policy;

Case 4: The NTSM control for a networked multi-manipulator system with the CD-UKF algorithm in the closed-loop and with the proposed neighbour selection policy.

Estimating Performance of CD-UKF Algorithm

We begin with **Case 1** without using the estimated states in the controller. In other words, it is assumed that all states are measurable, while estimations from the CD-UKF method are only for us to see its estimating quality. As can be seen in Fig.4.17, CD-UKF algorithm has a good ability to provide smooth average states out of the noisy measuring signals.

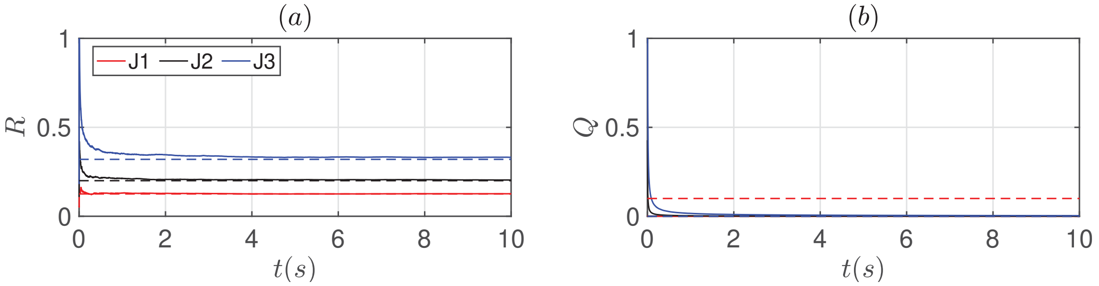


Figure 4.18: (a) The convergence of estimated measurement noise levels (solid lines) to their actual noise levels (dash lines). (b) Comparisons of estimated process noise levels to their actual noise levels (red dash lines).

As to the noise estimations, the measurement noise levels denoted by the covariance can be estimated accurately and promptly, as shown in Fig.4.18(a). However, the estimated process noises fail to converge to their true values (see Fig.4.18(b)). This is because the robust controller makes the system insensitive to the process noises. Therefore, the estimated process noises converge to zero regardless of the actual process noises.

Comparison Results

After confirming the good performance of the CD-UKF method, **Case 2** is considered. We simulated the closed-loop control scheme by feeding back the estimated states to the NTSM controller. Fig.4.19 shows the comparison results between **Case 1** and **Case 2**. Without loss of generality, when comparing tracking errors, we only present the results of the 3^{rd} manipulator along the x-axis as an example.

In both cases, we observe significant changes in the energy during the transient phase, and occasional fluctuations occur in the steady-state phase (see Fig.4.19(a)). In addition, the energy change of **Case 2** is slightly greater than that of **Case 1**. For instance, during $10 \sim 15$ s, the mechanical energy of **Case 2** experiences frequent oscillations with larger amplitudes than **Case 1**. This is caused by the jerky and oscillatory estimation of power signals (i.e., the velocity estimation in this case) using the CD-UKF approach. Consequently, the system tends to accumulate excessive energy.

Accordingly, as shown in Fig.4.19(b), more perturbations can be expected in the

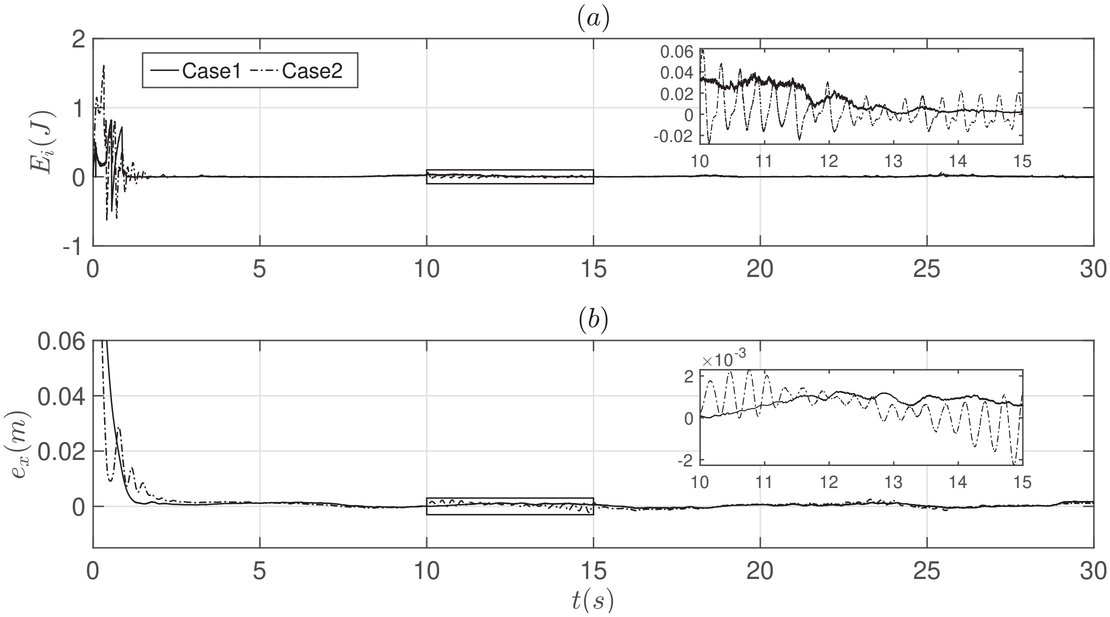


Figure 4.19: (a) The mechanical energy evolution and (b) tracking error (in x direction) of the 3rd manipulator in **Case 1** and **Case 2**.

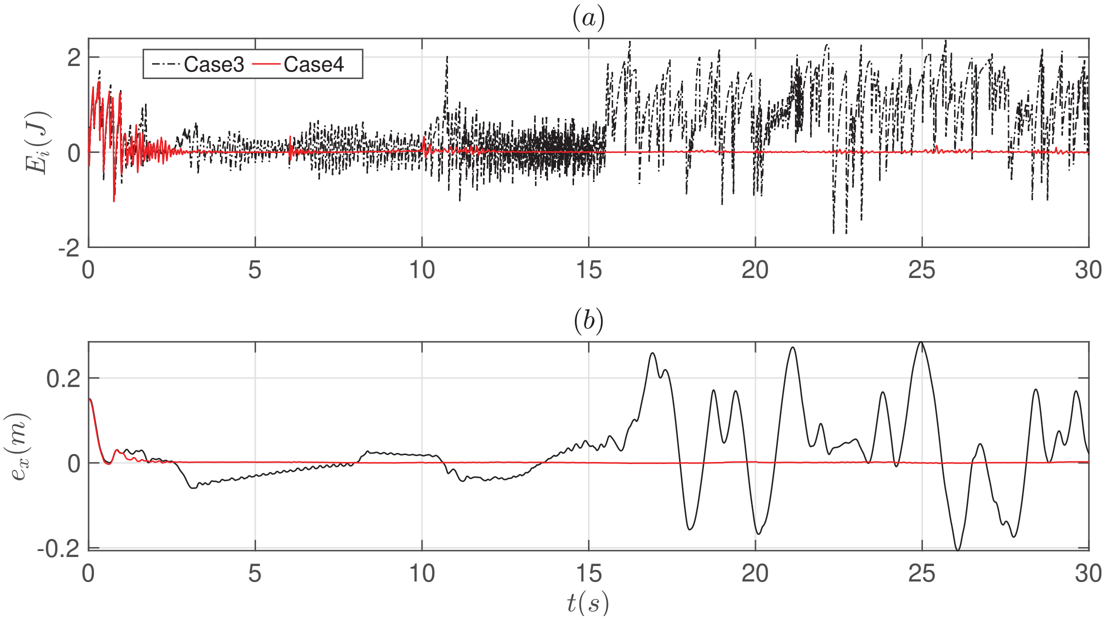


Figure 4.20: (a) The mechanical energy evolution and (b) tracking errors (in x direction) of the 3rd manipulator in **Case 3** and **Case 4**.

position signals of **Case 2**. Nevertheless, in many industrial applications, the tracking accuracy ($\leq 2 \times 10^{-3}$ m) as in **Case 2** is acceptable [102–104]. Therefore, for

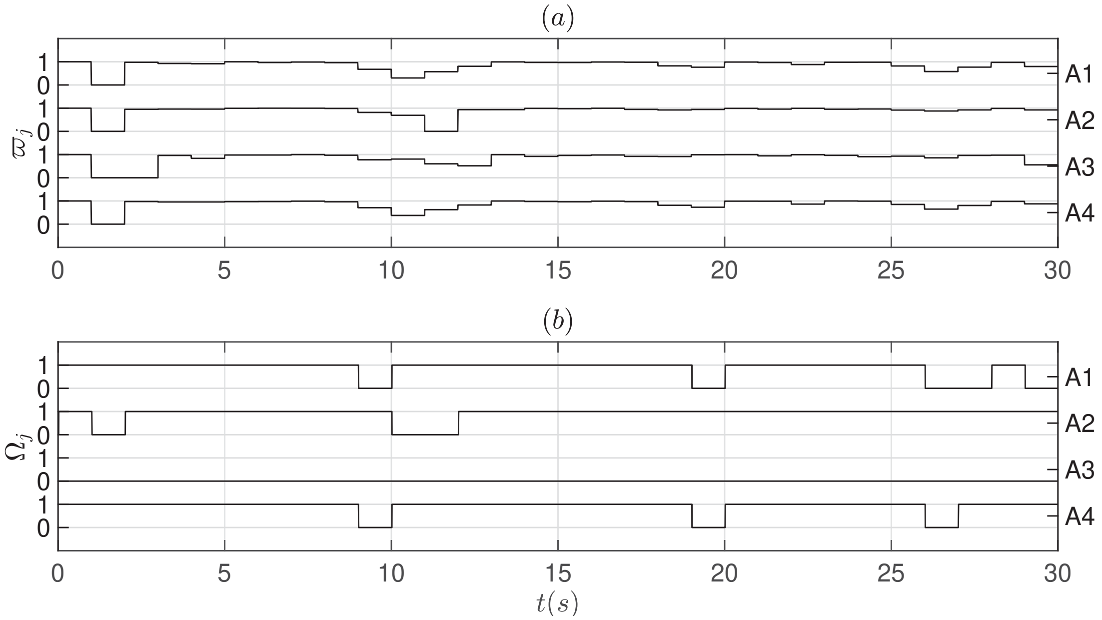


Figure 4.21: (a) Energy indexes of manipulator 1 ∼ 4 (denoted by A1 ∼ A4); (b) Selection indexes of agent 3 where $\Omega_j = 1$ indicates information of the neighbor j is used, and $\Omega_j = 0$ otherwise.

applications with a single manipulator, the NTSM control with the CD-UKF estimations can provide satisfactory performance. Next, we will examine the performance in a multi-manipulator system where the network delays and packet loss are involved.

Fig.4.20 and Fig.4.21 illustrate results of the 3^{rd} manipulator that has the same configurations as in **Case 1** and **Case 2**. Unfortunately, it can be seen from Fig.4.20 that the small perturbations in both energy and tracking error of **Case 2** have become significantly large in **Case 3**. The drastic amplification of oscillations and tracking errors in MASs results from multiple effects, including the intercommunications of the inaccurate and oscillatory states, network delays, packet loss, etc. Consequently, the overall performance degrades. As shown by the results of **Case 3** in Fig.4.20(b), the tracking error after 15 s is up to 0.2 m, which is far beyond desired performance.

Fig. 4.20 also presents the results of **Case 4** where the proposed neighbor selection policy is applied to address the issues in **Case 3**. As shown in Fig. 4.20, **Case 3** and **Case 4** have the same performance in the transient phase. Nonetheless, using the neighbour selection policy, the manipulator presents good tracking accuracy during the steady-state phase.

Fig.4.21 helps explain the reason why **Case 4** outperforms **Case 3**. Taking the time 9 ∼ 12 s as an example, we set the desired energy index $\varpi_{d,3} = 0.85$ for the 3rd manipulator (denoted by A3 as in Fig.4.21). When the energy indexes of its neighbours (i.e., A1, A2, and A4) decrease to be below the desired energy index of the 3rd manipulator, the 3rd manipulator decides to actively disconnect agent 1 and agent 4, as well as agent 2 subsequently. In **Case 4**, by cutting off the unnecessary interactions amongst the agents, the 3rd manipulator is less affected by the increasing tracking errors from its neighbours. In return, the 3rd manipulator is able to maintain its good tracking performance and serves as a good-performing neighbour for its connecting agents. This gives a healthier interaction amongst the manipulators compared to **Case 3**.

4.4 Concluding Remarks

The NTSM control method incorporating other techniques were developed for a leader-following cooperation system communicating in a weak topology in the presence of time delays. By using the mixed-type feedback signal and energy-based neighbour selections, the controllers are able to deal with issues of the large phase shift and the adverse neighbour influence, respectively. For each network-delayed agent, the performance that is measured by tracking error is significantly improved, when it is able to promptly react to the change of the reference signals, and, as well, receives less adverse influence from the noisy neighbours. The developed nonlinear controller has been theoretically and experimentally proved to be able to regulate the synchronization of multiple network-delayed manipulators, in the presence of practical constraints/challenges, such as network delays, weakly connected network, unknown dynamics and external noises.

Chapter 5

Adaptive NTSM Control with Time-varying Gains

Although a great deal of exact-model-based control policies have been theoretically proved to be able to achieve or to maintain a desired level of performance, from the practical point of view, these methods may be impracticable because the model parameters could be initially uncertain or vary during the operation. For example, the payload dynamics of a robot manipulator may be initially unknown and time-varying during the manipulation tasks. Therefore, to allow a control system to adjust itself to the unknown parameters, concepts of the adaptive control techniques have been developed.

In this chapter, we aim at developing the automated adaptation of the NTSM control method as discussed in Chapter 4. The purpose of the developed ANTSM is to address the deleterious effects caused by the initially unknown parameters, such as dynamic uncertainties, frictions, external noise bounds, and time-varying network delays, and, therefore, maintain the good synchronization performance[1].

5.1 Problem Description

In the leader-following framework, the leader is assumed as a given desired trajectory. Therefore the modelling of the leader as well as its control design will not be discussed. For the follower manipulators, they are modelled by the E-L representation as in (3.28) with the same variable definitions:

$$\bar{M}_i \ddot{\boldsymbol{x}}_i + \bar{C}_i \dot{\boldsymbol{x}}_i + \bar{\boldsymbol{g}}_i = \boldsymbol{u}_i + \boldsymbol{\rho}_i. \tag{5.1}$$

---

[1] This chapter was published as the work in [105], ©2019 AACC, ©2019 IEEE. Reprinted, with permission, from Henghua Shen and Ya-Jun Pan, "Adaptive Robust Control of Networked Multi-Manipulators with Time-Varying Delays", 2019 IEEE American Control Conference, July/2019, and as the work in [54], 0278-0046 ©2020 IEEE. Reprinted, with permission, from Henghua Shen and Ya-Jun Pan, "Tracking Synchronization Improvement of Networked Manipulators Using Novel Adaptive Non-Singular Terminal Sliding Mode Control", IEEE Transactions on Industrial Electronics, April/2020.

In (5.1), $\boldsymbol{\rho}_i$ compactly denotes the unknown components of the model parameters, including the model uncertainties, ΔM_i, ΔC_i, $\Delta \boldsymbol{g}_i$, the unknown external disturbances, $\boldsymbol{d}_i$, and the unknown friction, $\boldsymbol{f}_i^{\mathrm{f}}$, which can be expressed by

$$\boldsymbol{\rho}_i = \Delta M_i \ddot{\boldsymbol{x}}_i + \Delta C_i \dot{\boldsymbol{x}}_i + \Delta \boldsymbol{g}_i + \boldsymbol{d}_i + \boldsymbol{f}_i^{\mathrm{f}}. \tag{5.2}$$

Nonlinear control schemes that rely on the exact model information may not work as desired due to the adverse effect of the unknown parameters in $\boldsymbol{\rho}_i$. One can treat $\boldsymbol{\rho}_i$ as disturbances and use robust control approaches (e.g. SMC-based approaches) to solve the problem. However, the upper bound of $\boldsymbol{\rho}_i$ is required as a prior. To relax this requirement, we want to develop the adaptation in the NTSM approach to adjust estimates of the upper bounds. With the online estimation for the unknown parameters, the system is able to provide desired performance.

Another issue addressed in this chapter is the selection of NTSM control gain κ. Theoretically, the NTSM control method can well stabilize a nonlinear manipulator system as long as κ is a positive constant. However, in physical experiments, when a robot experiences significant static frictions, selecting a constant value of κ that guarantees the smooth convergence during the transient phase may not be able to provide high tracking accuracy during the steady-state phase, and vice versa. Therefore, we wish to find a time-varying control gain that is tunable in different phases to achieve a good trade-off between the smooth convergence and high steady-state tracking accuracy.

Control Objective

An adaptive NTSM controller will be designed to guarantee the position synchronization of the followers to a time-varying leader trajectory when the network communication is with unknown time-varying transmitting delays and weakly connectedness. In the presence of model uncertainties, unknown external noises, unknown frictions, and network delays, the network-wide objective is to achieve the leader-following synchronization described by $\|\boldsymbol{x}_i(t) - \boldsymbol{x}_0(t)\|_1 \leq \Delta_{\epsilon,i0}$ and $\|\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)\|_1 \leq \Delta_{\epsilon,ij}$. The bounds $\Delta_{\epsilon,i0} \geq 0$ and $\Delta_{\epsilon,ij} \geq 0$ are desired to be as small as possible. In addition, a time-varying ANTSM control gain is to design to guarantee the positions of the follower end effectors to smoothly converge to the leader from arbitrary initial positions, and, meanwhile, ensure the high tracking accuracy during the operation.

5.2 Adaptive NTSM Control with Time Varying Gains

In this section, an ANTSM control method in task space will be introduced with mixed-type feedback signals, three time-varying compensatory bounds, and a time-varying ANTSM control gain.

5.2.1 Mixed-Type Feedback Signals

As previously discussed in Chapter 4, The mixed-type signal can be designed as

$$\begin{aligned} \boldsymbol{e}_i^p &= \sum_{j=1}^{n} a_{ij}[\boldsymbol{x}_i(t-\bar{T}_{ij}) - \boldsymbol{x}_j(t-T_{ij}(t))] + b_i[\boldsymbol{x}_i(t-\bar{T}_{i0}) - \boldsymbol{x}_0(t-T_{i0}(t))], && (5.3) \\ \boldsymbol{e}_i^v &= \sum_{j=1}^{n} a_{ij}\Big[\dot{\boldsymbol{x}}_i(t) - \dot{\boldsymbol{x}}_j(t-T_{ij}(t))\Big] + b_i\Big[\dot{\boldsymbol{x}}_i(t) - \dot{\boldsymbol{x}}_0(t-T_{i0}(t))\Big], && (5.4) \end{aligned}$$

where $\boldsymbol{e}_i^p \in \mathbb{R}^{3\times 1}$ and $\boldsymbol{e}_i^v \in \mathbb{R}^{3\times 1}$ define the position error and velocity error, respectively. $T_{ij}(t)$ and $T_{i0}(t)$ are the time-varying network delays from agent j to i, and agent 0 to i, respectively. From the formulation in (3.31), we have $T_{ij}(t) = \bar{T}_{ij} + \delta_{ij}(t)$ and $T_{i0}(t) = \bar{T}_{i0} + \delta_{i0}(t)$, where $\bar{T}_{ij}$ and $\bar{T}_{i0}$ are nominal delays. It is assumed that $\bar{T} = \bar{T}_{ij} = \bar{T}_{i0}$ and they are constant and known. In the rest of this chapter, the dependence on t is omitted unless otherwise specified, e.g., $T_{i0} = T_{i0}(t)$, $T_{ij} = T_{ij}(t)$, $\dot{\delta}_{ij} = \dot{\delta}_{ij}(t)$, and $\dot{\delta}_{i0} = \dot{\delta}_{i0}(t)$. $\boldsymbol{x}_i(t-\bar{T}_{ij}) = \boldsymbol{x}_i(t-\bar{T}_{i0})$ are the positions with inexact self-delay.

Let $\boldsymbol{\varepsilon}_i^{sd} \in \mathbb{R}^{3\times 1}$ define the self-delayed velocity error that establish the relationship between the tracking errors in (5.3) and (5.4) as

$$\begin{aligned} \dot{\boldsymbol{e}}_i^p &= \boldsymbol{e}_i^v - \boldsymbol{\varepsilon}_i^{sd}, && (5.5) \\ \boldsymbol{\varepsilon}_i^{sd} &= \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)[\dot{\boldsymbol{x}}_i(t) - \dot{\boldsymbol{x}}_i(t-\bar{T})] - \sum_{j=1}^{n} a_{ij}\dot{\boldsymbol{x}}_j(t-T_{ij})\dot{\delta}_{ij} - b_i\dot{\boldsymbol{x}}_0(t-T_{i0})\dot{\delta}_{i0}. && (5.6) \end{aligned}$$

5.2.2 Compensatory Bounds

One of the advantages of the SMC methods is that any bounded undesired signals can be treated as disturbances and be well handled using the knowledge of the upper bounds. In what follows, three explicit error bounds are online estimated to compensate for the velocity and acceleration errors induced by the self-delay, acceleration estimation error, and time-varying network delays.

Self-Delay-Induced Velocity Error Bound

Assumed that the deviations $\delta_{i0}(t)$ and $\delta_{ij}(t)$ are changing sufficiently slowly such that $\|\dot{\delta}_{i0}(t)\|_1 \leq 1$ and $\|\dot{\delta}_{ij}(t)\|_1 \leq 1$ [50]. From (5.6), the compensatory bound for the self-delay-induced velocity error is defined by

$$\begin{aligned} \mathbb{B}_i^{sd} &= \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\|\dot{\boldsymbol{x}}_i(t) - \dot{\boldsymbol{x}}_i(t-\bar{T})\|_1 \\ &\quad + \sum_{j=1}^{n} a_{ij}\|\dot{\boldsymbol{x}}_j(t-T_{ij})\|_1 + b_i\|\dot{\boldsymbol{x}}_0(t-T_{i0})\|_1. \end{aligned} \tag{5.7}$$

Then, the inequality of $\|\boldsymbol{\varepsilon}_i^{sd}\|_1 \leq \mathbb{B}_i^{sd}$ always holds.

Acceleration Estimating Error Bound

In this chapter, the acceleration is estimated from the model in (5.1) with the known components, that is,

$$\hat{\ddot{\boldsymbol{x}}}_j(t-T_{ij}) = \bar{M}_j^{+}(t-T_{ij})\Big[\boldsymbol{u}_j(t-T_{ij}) - \bar{C}_j(t-T_{ij})\dot{\boldsymbol{x}}_j(t-T_{ij}) - \bar{\boldsymbol{g}}_j(t-T_{ij})\Big]. \tag{5.8}$$

As a result, comparing the estimating acceleration in (5.8) with the actual acceleration obtained from (5.1) gives the acceleration estimation error as

$$\boldsymbol{\varepsilon}_{ij}^{ae} = \ddot{\boldsymbol{x}}_j(t-T_{ij}) - \hat{\ddot{\boldsymbol{x}}}_j(t-T_{ij}) = \bar{M}_j^{+}(t-T_{ij})\boldsymbol{\rho}_j(t-T_{ij}). \tag{5.9}$$

Recall the bound of $\boldsymbol{\rho}_j(t-T_{ij})$ as in (3.29), we have,

$$\|\boldsymbol{\varepsilon}_{ij}^{ae}\|_1 \leq \|\bar{M}_j^{+}(t-T_{ij})\|_1\hat{\Theta}_j(t-T_{ij})\Phi_j(t-T_{ij}). \tag{5.10}$$

As the leader is assumed as a desired trajectory, no acceleration estimating error occurs in the leader manipulator. The network-wide acceleration estimation error only comes from neighbours, so the sum of the estimation errors is

$$\boldsymbol{\varepsilon}_i^{ae} = \sum_{j=1}^{n} a_{ij}[\ddot{\boldsymbol{x}}_j(t-T_{ij}) - \hat{\ddot{\boldsymbol{x}}}_j(t-T_{ij})] = \sum_{j=1}^{n} a_{ij}\boldsymbol{\varepsilon}_{ij}^{ae}. \tag{5.11}$$

Consider the inequalities in (5.10), let the compensatory bound for the acceleration estimating error be

$$\mathbb{B}_i^{ae} = \sum_{j=1}^{n} a_{ij} \|\bar{M}_j^{+}(t - T_{ij})\|_1 \hat{\Theta}_j(t - T_{ij})\Phi_j(t - T_{ij}), \tag{5.12}$$

such that we have $\|\boldsymbol{\varepsilon}_i^{ae}\|_1 \leq \mathbb{B}_i^{ae}$.

Delay-Variation-Induced Acceleration Error Bound

Another acceleration error is caused by the delay variation in the time-derivative operation of the time-delayed velocity signals. For example, the acceleration error without self-delay is

$$\boldsymbol{e}_i^a = \sum_{j=1}^{n} a_{ij} \Big[\ddot{\boldsymbol{x}}_i(t) - \ddot{\boldsymbol{x}}_j(t - T_{ij})\Big] + b_i \Big[\ddot{\boldsymbol{x}}_i(t) - \ddot{\boldsymbol{x}}_0(t - T_{i0})\Big],$$

while the acceleration error obtained by differentiating the velocity error in (5.4) is

$$\dot{\boldsymbol{e}}_i^v = \sum_{j=1}^{n} a_{ij} \Big[\ddot{\boldsymbol{x}}_i(t) - \ddot{\boldsymbol{x}}_j(t - T_{ij})(1 - \dot{T}_{ij})\Big] + b_i \Big[\ddot{\boldsymbol{x}}_i(t) - \ddot{\boldsymbol{x}}_0(t - T_{i0})(1 - \dot{T}_{i0})\Big].$$

Comparing $\boldsymbol{e}_i^a$ with $\dot{\boldsymbol{e}}_i^v$ gives rise to an acceleration error as

$$\boldsymbol{\varepsilon}_i^{tv} = \boldsymbol{e}_i^a - \dot{\boldsymbol{e}}_i^v = -\sum_{j=1}^{n} a_{ij} \ddot{\boldsymbol{x}}_j(t - T_{ij})\dot{T}_{ij} - b_i \ddot{\boldsymbol{x}}_0(t - T_{i0})\dot{T}_{i0}. \tag{5.13}$$

Because $\|\dot{T}_{ij}\| = \|\dot{\delta}_{ij}\| \leq 1$ and $\|\dot{T}_{i0}\| = \|\dot{\delta}_{i0}\| \leq 1$, the delay-variation-induced acceleration error is upper-bounded by

$$\|\boldsymbol{\varepsilon}_i^{tv}\|_1 \leq \sum_{j=1}^{n} a_{ij} \|\ddot{\boldsymbol{x}}_j(t - T_{ij})\|_1 + b_i \|\ddot{\boldsymbol{x}}_0(t - T_{i0})\|_1.$$

From (5.9) and (5.10), we have $\|\ddot{\boldsymbol{x}}_j(t-T_{ij})\|_1 \leq \|\ddot{\hat{\boldsymbol{x}}}_j(t-T_{ij})\|_1 + \|\bar{M}_j^{+}(t-T_{ij})\|_1 \hat{\Theta}_j(t-T_{ij})\Phi_j(t-T_{ij})$. Therefore, defining the compensatory bound as

$$\mathbb{B}_i^{tv} = \sum_{j=1}^{n} a_{ij} \|\ddot{\hat{\boldsymbol{x}}}_j(t - T_{ij})\|_1 + b_i \|\ddot{\boldsymbol{x}}_0(t - T_{i0})\|_1 + \mathbb{B}_i^{ae}, \tag{5.14}$$

ensures that the acceleration error caused by time-varying delays is upper-bounded by $\mathbb{B}_i^{tv}$, that is, $\|\boldsymbol{\varepsilon}_i^{tv}\|_1 \leq \mathbb{B}_i^{tv}$.

In the above, the three compensatory bounds in (5.7), (5.12), and (5.14) are all time-varying and can be calculated online from the available parameters.

5.2.3 Adaptive NTSM Control

Following the basics of the NTSM control design Section 3.5, the sliding surface of $\boldsymbol{s}_i \in \mathbb{R}^{3\times 1}$ for the control design is given by

$$\boldsymbol{s}_i = \boldsymbol{e}_i^p + \beta_i(\boldsymbol{e}_i^v)^{\alpha_i}, \tag{5.15}$$

and the followers' controller is then designed as

$$\begin{aligned}\boldsymbol{u}_i \;=\;& \bar{C}_i\dot{\boldsymbol{x}}_i + \bar{\boldsymbol{g}}_i + \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{-1}\bar{M}_i\bigg\{\frac{-(\boldsymbol{e}_i^v)^{(2-\alpha_i)}}{\alpha_i\beta_i} + \sum_{j=1}^{n} a_{ij}\ddot{\boldsymbol{x}}_j(t-T_{ij})\\ &+b_i\ddot{\boldsymbol{x}}_0(t-T_{i0}) - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\bar{M}_i^{M+}\hat{\Theta}_i\Phi_i\mathrm{sgn}(\boldsymbol{s}_i)\\ &-\Big[\mathbb{B}_i^{ae} + \mathbb{B}_i^{tv} + \frac{1}{\alpha_i\beta_i}\mathbb{B}_i^{sd}\mathrm{diag}((\boldsymbol{e}_i^v)^{(1-\alpha_i)})\Big]\mathrm{sgn}(\boldsymbol{s}_i) - \kappa_i\mathrm{sgn}(\boldsymbol{s}_i)\bigg\},\end{aligned} \tag{5.16}$$

where $\kappa_i > 0$ and $\bar{M}_i^{M+} \geq \|\bar{M}_i^{+}\|_1$. $(\boldsymbol{e}_i^v)^{(\alpha-1)}$ follows the same operation as in (4.18). $\hat{\Theta}_i$ is the estimate of Θ_i and obeys the following adaptive law in dealing with the unknown modeling parameters.

Theorem 5.1. *For the nonlinear manipulator model described in (5.1), the estimate of the unknown parameters, represented by $\tilde{\Theta}_i$, is repeatedly updated with the following adaptation law:*

$$\dot{\tilde{\Theta}}_i = -\alpha_i\beta_i\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\|\boldsymbol{s}_i^T\,diag[(\boldsymbol{e}_i^v)^{(\alpha_i-1)}]\|_1\bar{M}_i^{M+}\Phi_i, \tag{5.17}$$

so that the ANTSM controller in (5.16) is able to adjust itself and guarantee the tracking stability as well as the reachability of the sliding surface in (5.15) in a finite time.

The analysis of the control stability and the reachability of the sliding surface are provided in Appendix B.1.

However, successfully converging to the sliding surfaces is not necessarily equivalent to the fact that the tracking errors decay to zero or into bounds. Therefore, the following theorem describes the finite-time bound on the convergence of each follower towards the leader states.

Theorem 5.2. *For the control systems described by (5.1) and (5.16), the velocity error* $\boldsymbol{e}_i^v = \boldsymbol{0}$ *is not an attractor for* $\boldsymbol{s}_i \neq \boldsymbol{0}$ *and* $\tilde{\Theta}_i \neq 0$. *The tracking error* $\boldsymbol{e}_i^p$ *is bounded, and the convergence time is related to the control gains* α_i *and* β_i *as well as the initial states of the end effectors.*

The proof for Theorem 5.2 is presented in Appendix B.2.

5.2.4 Time-Varying Gain Design

As discussed in the proof of Theorem 5.1 (see Appendix B.1), the finite-time stability can be guaranteed as long as $\kappa_i > 0$ holds. However, when it comes to experiments, using the constant gain $\kappa_i > 0$ cannot always provide a smooth convergence to the desired trajectory during the transient phased and followed by a high accuracy motion tracking. In fact, due to the limitations of the maximum control inputs in many robots, high switching gains may result in high control energy that can cause some deleterious effects, such as large overshoots, the significant chattering, oscillations, or even the complete operation failure. On the other hand, using the smaller switching gains may not be able to counteract the static friction and thus significantly decrease the tracking accuracy in the steady-state phase. In what follows, a time-varying ANTSM control gain is introduced in a piecewise logistic form.

Lemma 5.1. *A logistic function (or logistic curve) is a function with a common "S" shape (sigmoid curve). It models the exponential growth over time with limits with the equation of*

$$f(t) = a + \frac{L}{1 + e^{-p(t - t_{sm})}}, \tag{5.18}$$

where a *denotes the lower asymptote,* L *denotes the curve's maximum value,* e *is the natural logarithm base (also known as Euler's Number),* p *represents the logistic growth rate, and* t_{sm} *denotes the sigmoid's midpoint.*

Lemma 5.2. *A logistic curves can be divided into three regions: an Initial Establishment Phase in which the growth is slow, followed by an Expansion Phase in which the value grows relatively quickly. Eventually the curve enters a long Entrenchment Phase where the output is close to its limit due to the intra-species competition.*

The following theorem and remarks describe the design of the time-varying control gain as well as effects of the parameters that characterize the control gain.

Theorem 5.3. *In the ANTSM controller in (5.16), the constant gain* κ_i *is replaced by a time-varying ANTSM gain* $\kappa_i(t)$ *that is given in a piecewise logistic form as*

$$\kappa_i(t) = \begin{cases} \kappa_i(0) + \frac{L_1}{1+e^{-p(t-t_{sm})}}, & 0 \leq t \leq 2t_{sm}, \\ \\ \frac{L_2}{1+e^{-p(t-1.8t_{sm})}}, & 2t_{sm} < t < \infty, \end{cases} \tag{5.19}$$

such that in physical experiments, the ANTSM controller enables the end effectors' position to smoothly converge to the desired trajectory from any initial positions. Meanwhile, the high tracking accuracy can be achieved after the convergence. $\kappa_i(0)$ *is the initial control gain. Definitions related to the logistic function are provided in (5.18).*

Remark 5.1. *In design of the ANTSM control gain, the nonlinear logistic function provides adjustable parameters, including the natural logarithm base(e), the logistic growth rate (p), and sigmoid's midpoint* (t_{sm})*, to manage the transition from the low gain* ($\kappa_i(0)$) *to the high gain (L). By properly selecting the adjustable parameters, the overshoot/undershoot can be significantly reduced when switching the small gain to the high gain.*

In (5.19), $2tsm$ and $-1.8tsm$ are determined by experience. This discontinuity is designed to provide a sudden input impulse at the end of the transient phase to "break" the "sticky" static friction. As a result, higher steady-state tracking accuracy can be achieved. The duration of the *Initial Establishment Phase* that is denoted as the *establishment time* T_{etm} is mainly determined by the parameters of e, p and t_{sm}. The end effector position profile may present a compound waveform by using the time-varying gain in (5.19), including a pre-transition aberration region and a post-transition aberration region.

The pre-transition overshoot/undershoot is associated with the initial control gain $\kappa_i(0)$ and the maximum value L_1 and the duration of the pre-transition aberration region of the position profile is affected by the establishment time T_{etm}. In the post-transition aberration region, the maximum value L_2 determines the overshoot/undershoot. Definitions of different components in a waveform are given in Appendix D.

Table 5.1: The oscillation and settling time with different constant κ_i.

	κ_i					
	1	2	4	6	8	10
OS (%)	1.25	6.85	61.70	82.92	85.55	84.30
US (%)	0	0	9.49	70.90	102.01	126.29
ST (s)	0.169	0.157	0.251	0.327	0.414	0.562

OS: Overshoot; US: Undershoot; ST: Settling Time

5.3 Discussions on Control Gain κ_i

In this section, we will discuss the constraint of using the constant switching gain κ_i and the benefit of using the proposed time-varying control gain $\kappa_i(t)$. The effects of multiple variables in (5.19) on the tracking performance were experimentally tested by a Phantom Omni haptic device that is shown in Fig.3.4 (b).

5.3.1 Results of Constant Gain Control

As previously discussed, using the constant control gain κ_i, one may either achieve high tracking accuracy but significant overshoot/undershoot with a large control gain, or a smooth convergence but low tracking accuracy with a smaller κ_i. This constraint is illustrated in Fig. 5.1. It shows that the increasing κ_i value improves the tracking accuracy during the steady-state but leads to an increasingly significant oscillation during the transient phase. The overshoot/undershoot measurements during the transition phase with respect to κ_i are given in Table. 5.1. Choosing the control gain $\kappa_i = 10$ provides a high tracking accuracy, i.e., $Ex_{ss} = 0.0018\ m$ (as shown in Fig. 5.1(b)), and, however, results in a high overshoot/undershoot, i.e., $OS = 84.30\%$ and $US = 126.29\%$. In contrast, the lower control gain, e.g., $\kappa_i = 1$, gives a very smooth transition, but the tracking error grows to $Ex_{ss} = 0.008\ m$. Besides, as seen in Table. 5.1, the greater value of κ_i results in a longer settling time. The results well illustrate the limitation of using constant control gains κ_i.

5.3.2 Results of the Time-varying Gain control

Before giving detailed discussions on the time-varying control gain $\kappa_i(t)$, Fig.5.2 (a) depicts a sample result of using $\kappa_i(t)$ comparing with those of using the constant

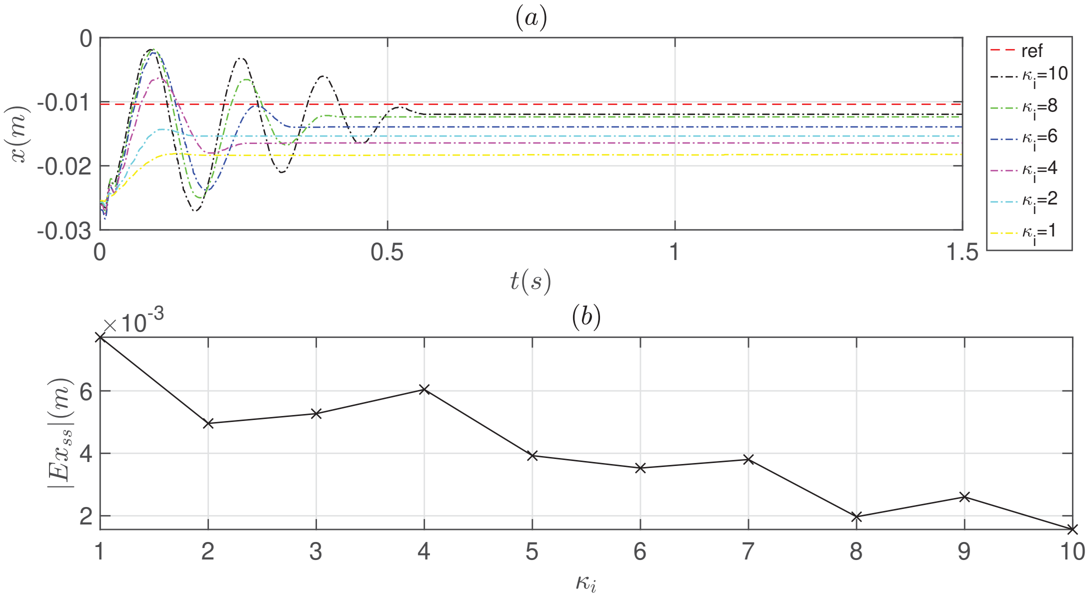


Figure 5.1: (a) Position profiles with different fixed κ_i; (b) The relation between κ_i and steady-state error Ex_{ss} with $E_x(0) = 0.03\ m$.

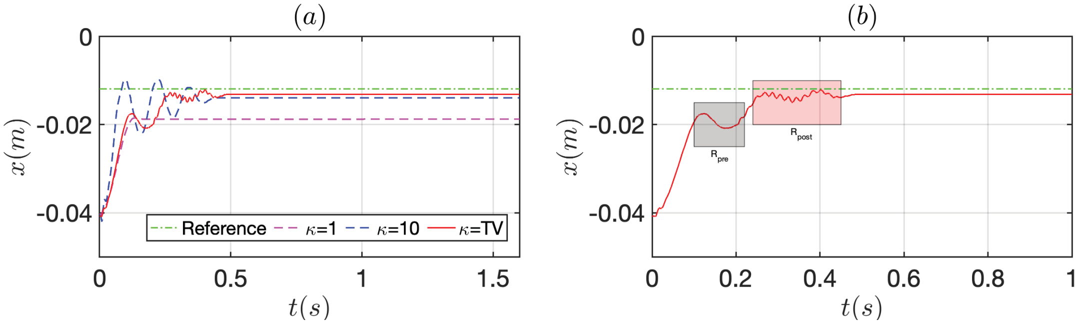


Figure 5.2: (a) Comparison of the tracking results using time-varying gain and constant gains; (b) Two aberration regions: R_{pre} and R_{post}.

gains ($\kappa_i = 1$ and $\kappa_i = 10$, respectively). The logistic parameters for the time-varying control gain were set to $\kappa_0 = 1.8$, $t_{sm} = 0.13$, and $L_1 = 5$ and $L_2 = 17$. As expected, using the time-varying control gain $\kappa_i(t)$ greatly improves the steady-state accuracy compared with the result of using $\kappa_i = 1$, and, meanwhile, reduces the oscillation in the transient phase, compared with the result of using $\kappa_i = 10$. However, as shown in Fig.5.2 (b), by using the time-varying gain in the ANTSM controller, the position profile presents a compound waveform, including a pre-transition aberration region (R_{pre}) and a post-transition aberration region (R_{post}). Some small oscillations can be observed too. Therefore, in the following experiments, we aim at finding solutions

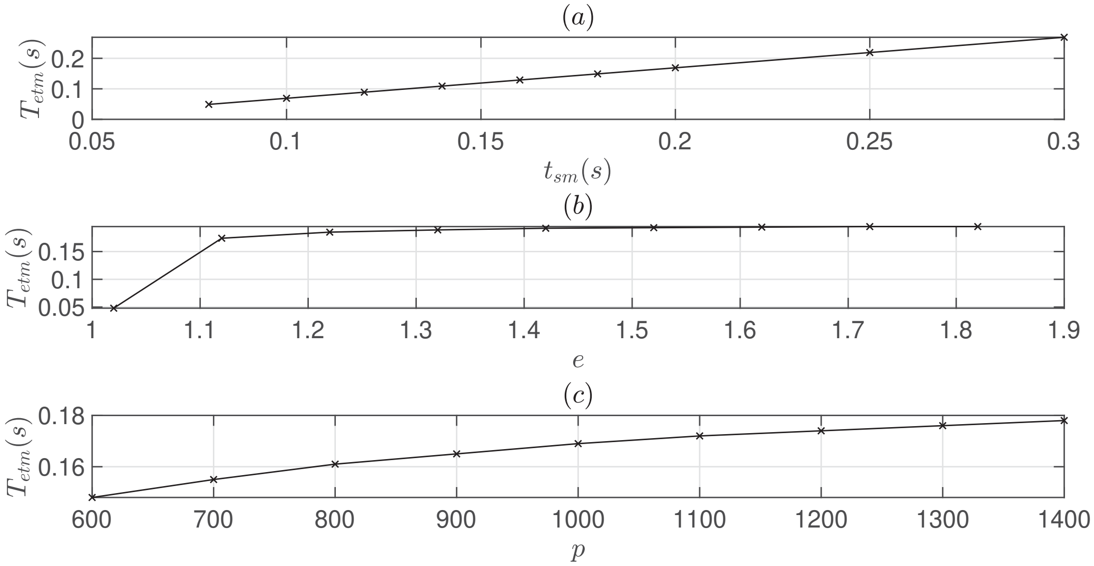


Figure 5.3: The effects of the Sigmoid midpoint t_{sm}, logarithm base e, and growth rate p on the establishment time T_{etm}.

to improve the tracking performance in terms of the settling time, oscillation and steady-state error by empirically examining effects of the establishment time T_{etm}, lower asymptote $\kappa_i(0)$, and the maximum values: L_1 and L_2 during the two aberration regions, R_{pre} and R_{post}.

Settling Time

Fig.5.3 demonstrates the change of the establishment time T_{etm} with respect to three logistic parameters: e, p and t_{sm}. Fig.5.3 (a) shows that the establishment duration linearly increases with the increasing Sigmoid's midpoint t_{sm}. The natural base e has less effect on T_{etm}, especially when $e \geq 1.1$ (see Fig.5.3 (b)). In Fig.5.3 (c), T_{etm} changes exponentially with respect to the growth rate p. Therefore, we can tune the Sigmoid's midpoint t_{sm} and/or growth rate p to adjust the establishment time T_{etm},

Since the transition features mostly present in the pre-transition aberration region, the control gain $\kappa_i(t)$ was given by a half-logistic function during the time $0 \sim t_{sm}$, that is,

$$\kappa_i(t) = \begin{cases} \kappa_i(0) + \frac{L_1}{1+e^{-p(t-t_{sm})}}, & 0 \leq t \leq t_{sm}, \\ \kappa_i(0) + \frac{L_1}{2}, & t_{sm} < t < \infty. \end{cases} \tag{5.20}$$

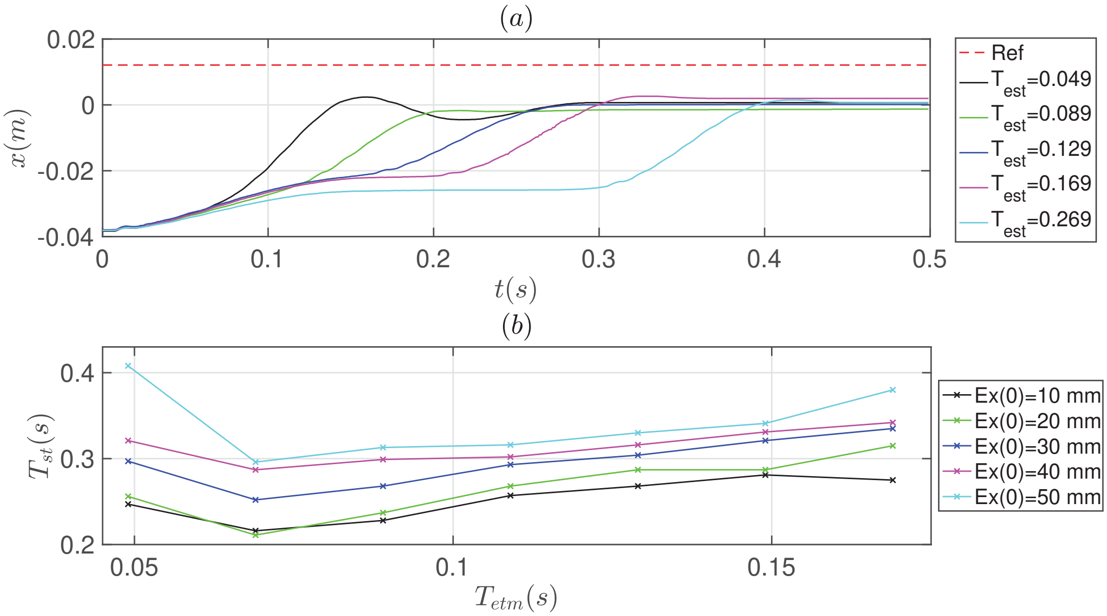


Figure 5.4: (a) The effect of the establishment time on the duration of the pre-transition aberration region; (b) Relation of establishment time and the settling time under different initial errors.

The initial error was set to $E_x(0) = 40\ mm$ and the control gain was selected with $\kappa_i(0) = 1.1$, $e = 1.1$, $p = 1000$, $t_{sm} = 0.13$, and $L_1 = 15$. The transition characteristics are then investigated in terms of κ_0, L_1, and the establishment time that is determined by e, p, and t_{sm} as in Fig.5.3.

Fig.5.4 (a) illustrates that the pre-transition aberration region appears when $T_{etm} \geq 0.129\ s$ and the duration of R_{pre} tends to become longer as T_{etm} keeps increasing. This finding is consistent with the results shown in Fig.5.4 (b) where the longer T_{etm} tends to increase the settling time. For example, given $E_x(0) = 50\ mm$, when $T_{etm} = 0.069\ sec$, the settling time is $T_{st} = 0.296\ sec$, while the settling time increases to $T_{st} = 0.38\ sec$ when $T_{etm} = 0.169\ sec$.

Oscillations

As oscillations may occur in both the pre-transition aberration region (R_{pre}) and the post-transition aberration region (R_{post}), reasons that cause oscillations in the two regions are examined as follows.

Oscillations in $\boldsymbol{R_{pre}}$: Experimental results of using the constant control gain

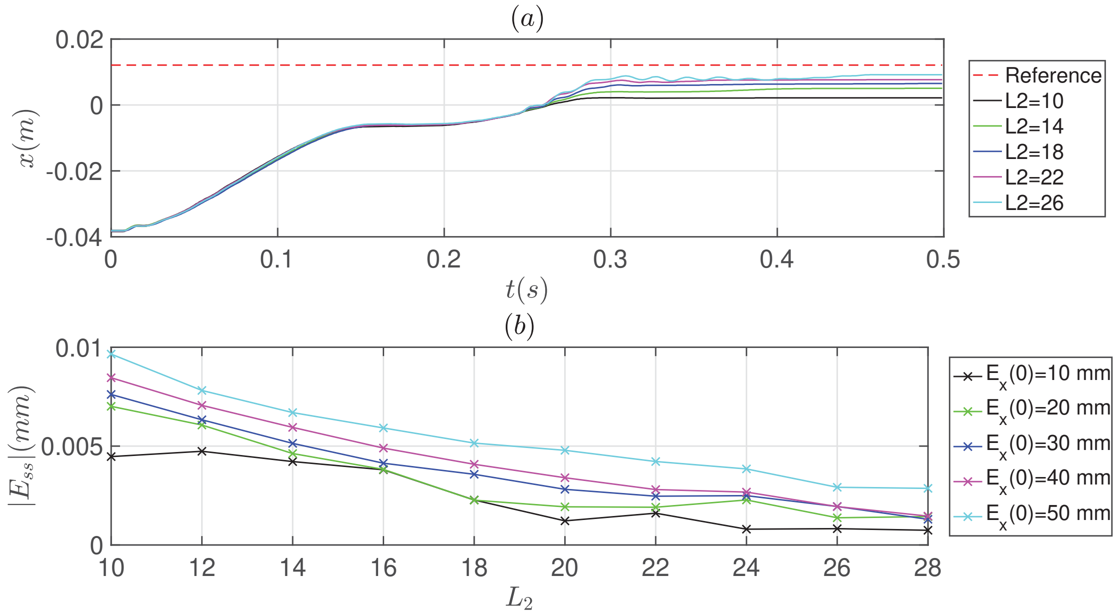


Figure 5.5: (a) Tracking results with different maximum values L_2; (b) Relationship between L_2 and the steady-state error.

suggest that a sufficiently large initial control gain $\kappa_i(0)$ can cause the oscillations in R_{pre}. Using the half-logistic control gain in (5.20), Table.5.2 presents the oscillations indicated by the overshoot/undershoot with respect to the initial control gain $\kappa_i(0)$ and under different initial errors from 10 mm to 50 mm. As expected, selecting a larger $\kappa_i(0)$ results in oscillations in R_{pre}. Similarly, a larger initial error can cause significant overshoot/undershoot. Therefore, it is advised to select a smaller $\kappa_i(0)$ when the initial errors are considerably large.

Oscillations in $\boldsymbol{R_{post}}$: To further investigate the effects of the maximum values L_1 and L_2 in the post-transition aberration region, the full-logistic control gain as in (5.19) was used and the initial control gain was $\kappa_i(0) = 1.1$. Table.5.3 and Table.5.4 present the testing results of the oscillations with respect to L_1 and L_2, respectively, during R_{post}.

When the initial error is small (e.g., $E_x(0) = 10$ mm), selecting a higher value of L_1 is more likely to excite significant oscillations, while for larger initial error (e.g., $E_x(0) = 50$ mm), the oscillation is less significantly affected by L_1. However, using a lower L_1 may not be able to provide sufficient torque input to generate a post-transition aberration region, which is necessary for improving the steady-state

Table 5.2: The oscillation (unit: %) subject to the intial switching gain $\kappa_i(0)$.

$\kappa_i(0)$	Initial Errors $E_x(0)$									
	10 mm		20 mm		30 mm		40 mm		50 mm	
	OS_{pre}	US_{pre}	OS_{pre}	US_{pre}	OS_{pre}	US_{pre}	OS_{pre}	US_{pre}	OS_{pre}	US_{pre}
0.2	0	0	0	0	0	0	0	0	0	0
0.5	0	0	0	0	0	0	0	0	0	0
0.8	0	0	0	0	0	0	0	0	0	0
1.1	0	0	0.676	0	0.765	0	0.430	0	0.297	0
1.4	0.6725	0	1.094	0	0.735	0	1.491	0	2.035	0
1.7	2.1487	0	1.014	0	3.133	0	4.346	0	10.296	0
2.0	1.4611	0	1.974	0	5.324	0	11.430	0	16.783	0
2.3	2.3683	0	6.477	0	7.840	0	20.273	0	28.649	-0.5812
2.6	3.6984	0	9.981	0	18.628	0	30.456	0	38.264	-1.6416

Table 5.3: The oscillation (unit: %) subject to the maximum value L_1.

L_1	Initial Errors $E_x(0)$									
	10 mm		20 mm		30 mm		40 mm		50 mm	
	OS_{post}	US_{post}	OS_{post}	US_{post}	OS_{post}	US_{post}	OS_{post}	US_{post}	OS_{post}	US_{post}
6	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	2.226	0	0.321	0	2.000	-0.906
10	0.464	0	2.052	0	11.722	-1.086	8.158	-2.832	6.506	-13.164
12	16.680	0	12.957	-0.327	11.667	-16.854	10.988	-16.009	5.048	-11.434
14	42.193	-7.436	15.590	-12.519	14.085	-25.674	7.304	-16.774	4.229	-10.221

accuracy. To investigate the effect of the maximum value, L_2, in R_{post}, $L_1 = 6$ was selected to guarantee the smoothness of the transition. Table.5.4 shows the results of the overshoot/undershoot in R_{post} as L_2 increases from 10 to 28. It shows that, with smaller initial errors, significant oscillations can be expected when L_2 increases up to 28. Whereas the oscillation is smaller when the initial errors are large, e.g., $E_x(0) = 50$ mm, although higher L_2 values were chosen. Therefore, the selection of L_2 is also subject to the initial distance between the follower end effectors and the reference trajectories.

Table 5.4: The oscillation (unit: %) subject to the maximum value L_2.

L_2	Initial Errors $E_x(0)$									
	10 mm		20 mm		30 mm		40 mm		50 mm	
	OS_{post}	US_{post}	OS_{post}	US_{post}	OS_{post}	US_{post}	OS_{post}	US_{post}	OS_{post}	US_{post}
10	0	0	-0.123	-0.951	0	-0.231	0	-0.162	0	-0.124
12	0	0	-0.271	-1.112	0	-0.433	-0.025	-0.183	-0.058	-0.295
14	0	-0.136	0	-0.386	0	-0.212	0	-0.151	0	-0.242
16	-0.269	-1.116	0	0	0	-0.201	-0.104	-0.104	0	-0.113
18	0	0	0	0	0	0	-0.017	-0.017	0	-0.111
20	2.967	-0.080	0.395	-0.184	0.224	-0.283	0	0	0	-0.110
22	19.849	-3.246	6.670	-0.202	2.228	-0.178	0.430	-0.290	0	-0.108
24	21.380	-23.281	13.272	-2.527	3.944	-1.801	1.375	-1.271	0	0
26	31.227	-37.855	11.972	-16.005	5.134	-7.218	1.364	-4.609	0	0
28	34.549	-52.655	16.983	-22.855	8.706	-13.107	3.607	-8.269	0	0

Steady-State Error

In the above, the establishment time T_{etm}, initial control gain $\kappa_i(0)$, and maximum value L_1 post more influence on the performance of the pre-transition aberration region R_{pre}, while performance of the post-transition aberration region R_{post} is primarily associated with the maximum value L_2. By properly selecting T_{etm}, $\kappa_i(0)$ and L_1 to achieve a smooth R_{pre} waveform, we investigate the effect of L_2 on the steady-state accuracy in R_{post}. Fig.5.5 (a) depicts the effects of L_2 values on the steady-state tracking accuracy. Using a higher L_2 value can improve the tracking accuracy, which can be seen from Fig.5.5 (b) where the tracking accuracy, represented by the absolute static error $|E_{ss}|$, improves with the increasing L_2 regardless the initial tracking error. In addition, with the same L_2 value, the steady-state error is lower when the initial position error is smaller. Therefore, for applications with large initial errors, it is advised to select a greater value of L_2.

Designing Rules of the Time-varying Control Gain

Based on the above analysis, guidelines to design the time-varying control gain are summarized as follows.

The effects of parameters t_{sm}, e and p can be characterized by the establishment time T_{etm} of the initial establishment phase. It is recommended to fix the values of e and p (e.g., $e = 1.1$ and $p = 1000$), such that the length of the establishment time T_{etm}

can be linearly changed by arranging location of the Sigmoid's point t_{sm}. T_{etm} is the primary factor in tuning the length of settling time. As T_{etm} has less influence on the steady-state accuracy in R_{pre}, users are suggested to increase T_{etm} from a relatively small value until obtaining an optimal value when it is right before the pre-transition aberration region appears. This optimal value allows a short settling time and avoids the significant overshoot/undershoot.

In the selection of $\kappa_i(0)$, according to the results in Table.5.2, it is recommended to select $\kappa_i(0) \leq 1.4$ to avoid significant oscillations regardless the amplitude of initial errors.

The selection of the maximum values, L_1 and L_2, is subject to the initial errors. To repeat experiments without concerning the initial error, users are suggested to select a smaller value of L_1 (e.g., $L_1 < 8$ in this experiment) so that less significant oscillations can be observed. For L_2, since a larger L_2 gives better steady-state accuracy but also excites significant oscillations in R_{post} as in Table.5.4, a trade-off between oscillations and the steady-state accuracy is required when selecting L_2 values. For example, for any initial error in this experiment, $L_2 = 20$ was selected as it allows the relatively small oscillation and drives the tracking error within 5×10^{-3} m.

The above guidelines help users have a quick control gain tuning to obtain optimal performance. Note that tuning results of the logistic parameters may change a bit due to the individual use of devices with different nonlinearities.

5.4 Illustrative Examples

In this section, the proposed methods were applied to the synchronization control of a multi-manipulator system through numerical simulations as well as experiments. In what follows, we will demonstrate two illustrative examples using 2-DOF planar manipulators and Phantom Omni haptic devices, respectively.

5.4.1 Simulation Results

In the numerical simulations, a group of identical 2-DOF ($k = 2$) manipulators as modelled in Section 3.3.3 were used to test the proposed control approach with constant control gain. Different initial joint positions of manipulators were selected so that each end effector had a different initial pose, that is (unit: rad), $\boldsymbol{q}_1(0) =$

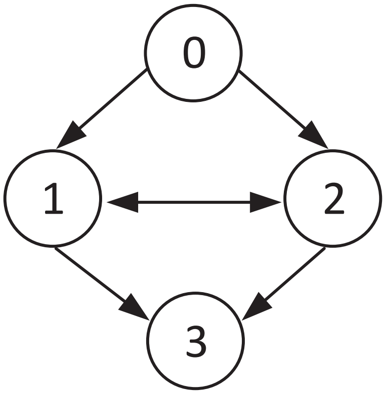


Figure 5.6: Weakly connected topology of the leader-following manipulator system.

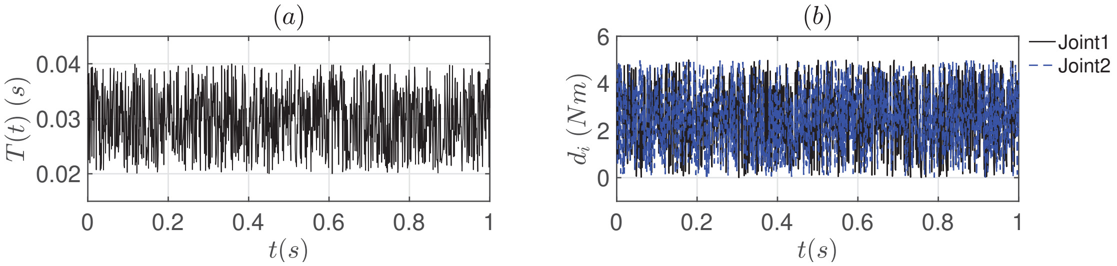


Figure 5.7: (a) The time-varying network delays, and (b) the external torque disturbances.

$[1.500 \quad 2.000]^T$, $\boldsymbol{q}_2(0) = [1.000 \quad 1.000]^T$, $\boldsymbol{q}_3(0) = [0.500 \quad 0.500]^T$. The initial joint velocities were 0 for all manipulators.

In the simulation, the graph contains one leader and three followers, as shown in Fig.5.6 where the followers are weakly connected, and the leader is globally reachable. The leader was given as a time-varying trajectory, i.e., sinusoidal signals of $\boldsymbol{x}_0 = J_0[0.800 \quad 0.800\sin(\pi t/40)]^T \ m$. The same control gains were selected as: $\alpha = 7/5$, $\beta = 5$, and $\kappa = 10$ for the follower manipulators.

The network delays in the simulation were varying with a nominal delay of $\bar{T} = 0.03\ s$ [88]. The network delays and the external disturbance are depicted in Fig.5.7. Assume that the parameter uncertainties were half of the true values, that is, $\Delta M_i = \frac{1}{2}M_i$, $\Delta C_i = \frac{1}{2}C_i$, and $\Delta \boldsymbol{g}_i = \frac{1}{2}\boldsymbol{g}_i$. The friction was given with the coefficients of $\gamma_i^1 = 3, \gamma_i^2 = 6, \gamma_i^3 = 1, \gamma_i^4 = 2, \gamma_i^5 = 4, \gamma_i^6 = 1$.

The position tracking in Fig.5.8 (a) and (b) demonstrates that the synchronization of the followers to the leader can be achieved in a finite time (about 10 s). During the steady-state phase, the tracking errors are small and bounded in $[-4.294,\ 5.516] \times$

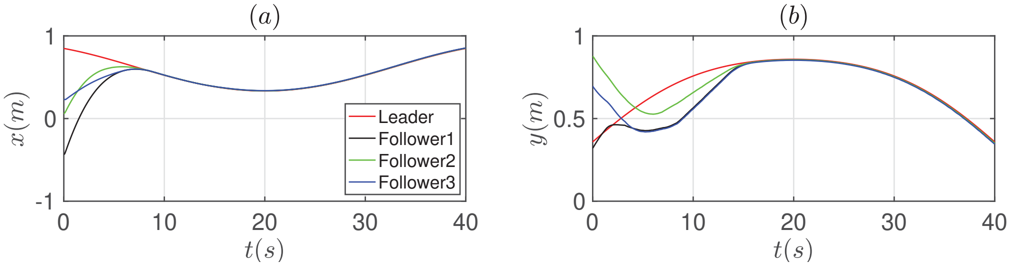


Figure 5.8: Position tracking of the end effectors along (a) x direction and (b) y direction.

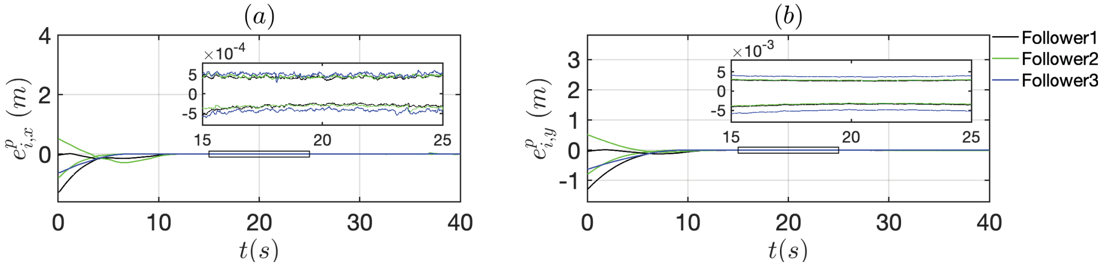


Figure 5.9: Position tracking errors between the followers and the leader by using (a) sign function, and (b) saturation function in the follower controllers.

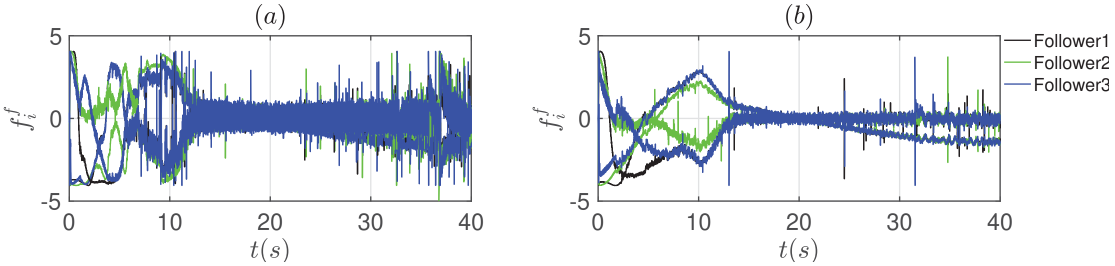


Figure 5.10: The resultant friction of follower manipulators using (a) sign function, and (b) saturation function in the controller.

10^{-4} m, as depicted in Fig.5.9 (a).

However, as shown in Fig.5.10 (a), the frictions profile of the followers present the significant chatter. This is caused by the sign function used in the SMC method. In practice, saturation function as in (4.22) is used instead to reduce the chattering effect. Fig.5.10 (b) depicts the chattering reduction by using the saturation function. However, using the saturation function in the control design sacrifices a certain amount of accuracy. This is illustrated by comparing the results between Fig.5.9 (a) and (b). Using the saturation function in the control design, the tracking error bounds increase to $[-4.888,\ 3.736] \times 10^{-3}$ m, which is approximately ten times higher

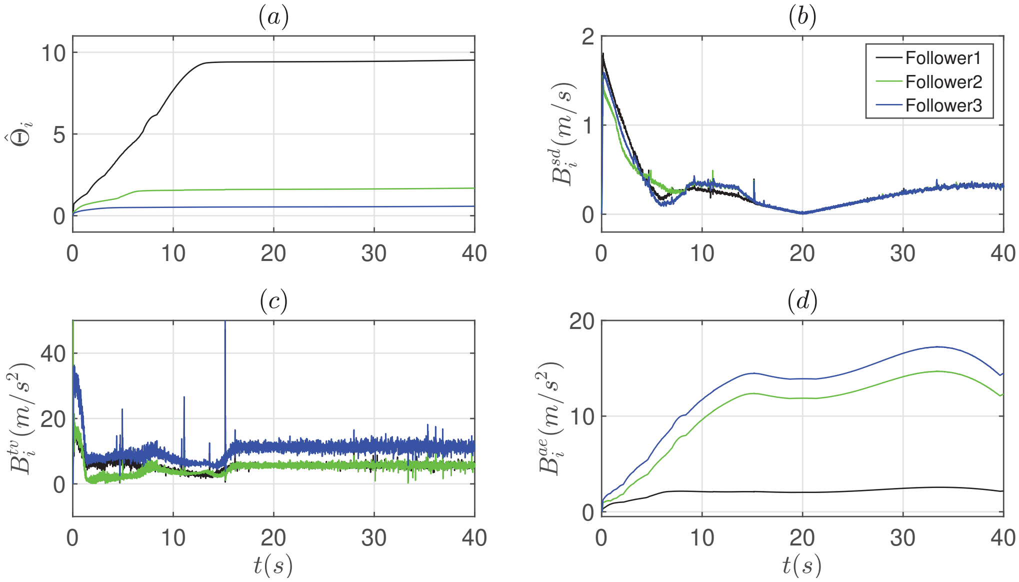


Figure 5.11: (a) The adaption of the unknown parameter $\hat{\Theta}_i$; The compensatory bounds for (b) for velocity error caused by self-delay; (c) for acceleration error caused by time-varying delays; (d) for acceleration estimating error.

than that in Fig.5.9 (a). Therefore, by adjusting the value of Δ, we can find a proper trade-off between the chattering effect and the tracking accuracy to achieve desired performance.

Fig.5.11 (a) illustrates that the ANTSM control approach provides a fast estimation of the unknown parameter $\hat{\Theta}_i$ that consists of all unknown upper bounds, $k_{0,i}^1 \sim k_{0,i}^6$. As expected, $\hat{\Theta}_i$ converges to constants in a finite time, about 10 s. Notice that, depending on the individual requirements, the convergence time is adjustable by properly setting the initial position error and the control gains. During the operation, the three compensatory bounds are expected to vary within certain reasonable ranges, and this is verified in Fig.5.11 (b)-(d).

5.4.2 Experimental Results

In the experimental tests, the control approach with the time-varying control gains $\kappa_i(t)$ in (5.19) was validated. A virtual leader and two followers running by the Phantom Omni haptic devices were used, as shown in Fig.5.12. The topology is denoted by $A = [0\ 0; 1\ 0]$ and $b = [1\ 0]^T$. Two control parameters were set to $\alpha = 7/5$, $\beta = 1$, and for the time-varying switching gain $\kappa_i(t)$, the following parameters were

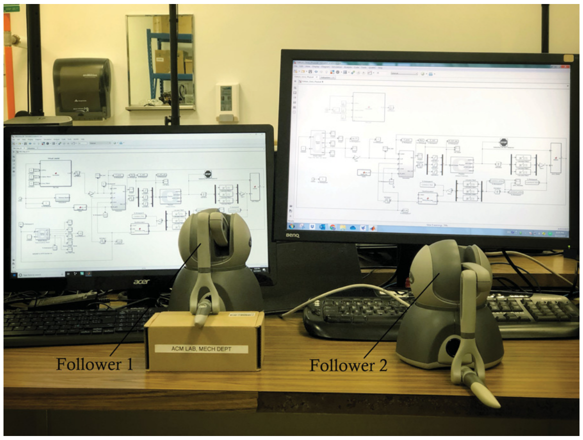


Figure 5.12: Testbed of the networked leader-following system with a pair of Phantom Omni haptic devices.

selected: $\kappa(0) = 1.1$, $t_{sm} = 0.13$, $L_1 = 2.0$ and $L_2 = 18$. Notice that the values for the control gains were selected from the experimental testing with the Phantom Omni haptic devices. For other robotic devices, those parameters may need to be reselected based on the rules in Section 5.3.2.

The Wireless LAN communication was implemented with the average one-way delay of $\bar{T} = 6.3 \times 10^{-3}$ sec (see Fig.3.7 (c) and (d)). The time-varying prescribed reference trajectories were recorded from operating a Phantom Omni haptic device. Both arms were then controlled to repeat the recorded trajectories. As the Omni devices only give position measurements, a first-order low-pass filter, $G(s) = \frac{1}{0.09s+1}$, was used to reduce the high frequency of the velocity and acceleration signals that were obtained by the time-differentiation.

The tracking results are given in Fig.5.13 (a)-(c), showing that the position of the two followers along x, y, z directions synchronize to the leader trajectory smoothly. As expected, small and bounded tracking errors during the steady-state phase can be achieved. Fig.5.13 (d)-(f) show that the steady-state errors are all approximately within $[-6,\ 6] \times 10^{-3}$ m.

To protect the devices from overloading, a saturation block is used to limit the control input signal in joint space. The upper and lower saturation values were $\boldsymbol{\tau}_{m,s} \in [-0.200\ 0.200]$ Nm. Fig.5.14 (a) and (b) respectively present the control input of the two followers. The proposed time-varying gain helps prevent the persistent over-limit control inputs during the transient phase. The chattering effect in the

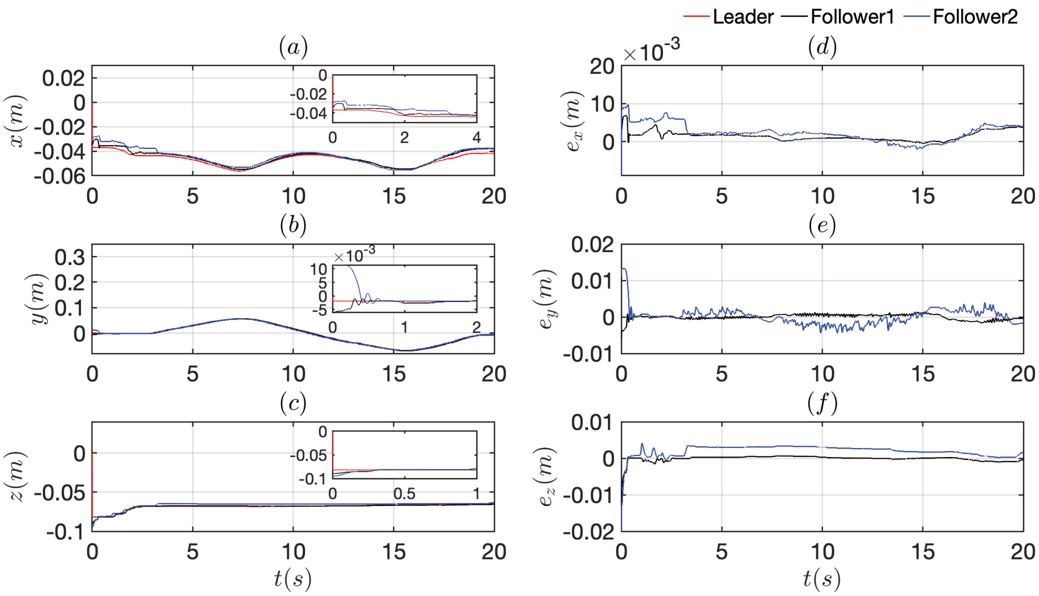


Figure 5.13: Position tracking and the tracking errors of the leader-following system with time-varying human-commanded reference signals.

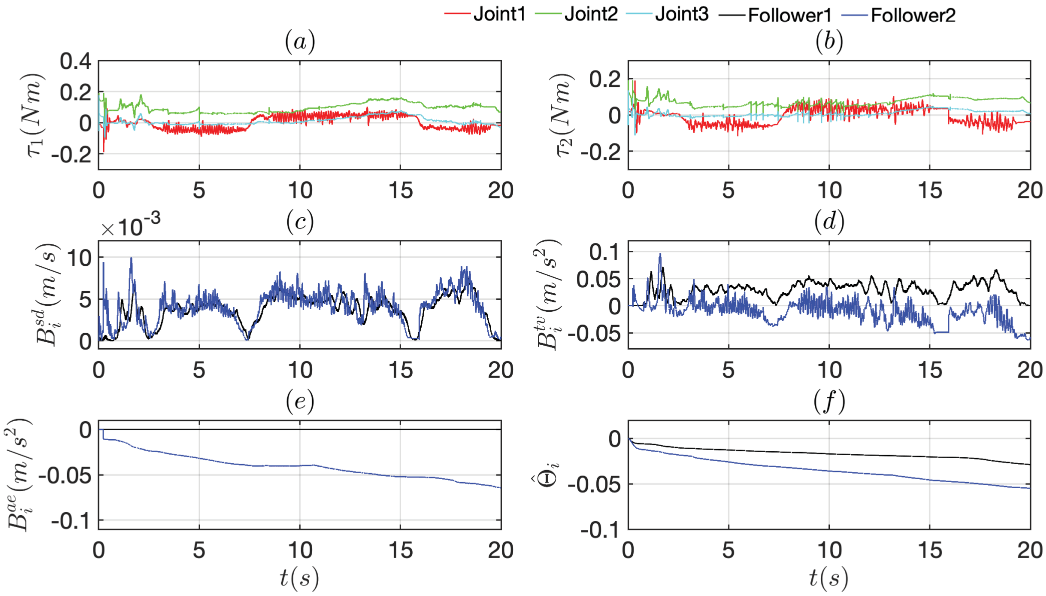


Figure 5.14: Filtered control inputs for (a) the follower 1 and (b) the follower 2; The compensatory bounds: (c) $\mathbb{B}^{sd}$, (d) $\mathbb{B}^{tv}$ and (e) $\mathbb{B}^{ae}$; (f) Estimation parameter $\hat{\Theta}$.

control input signals is reduced by using the saturation function, although a certain level of chattering effect can still be observed.

Fig.5.14 (c) and (d) depict that during the steady-state phase, the compensatory bounds $\mathbb{B}_s^{sd}$ and $\mathbb{B}^{tv}$ are bounded within a small range, i.e., $\mathbb{B}^{sd} \leq 0.004\ m/s$ and

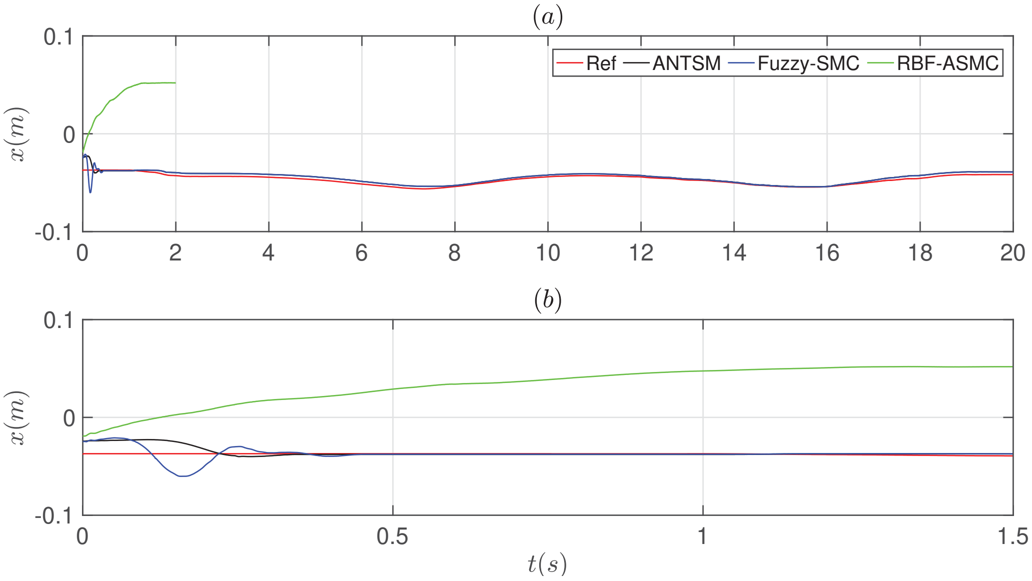


Figure 5.15: (a) Tracking performance comparison amongst Fuzzy-SMC method, RBFNN-ASMC method, and the proposed ANTSM approach (Taking x-axis as an example); (b) Zoomed-in plot during $t = 0 - 1.5$ *sec*.

$\mathbb{B}^{tv} \leq 0.150\ m/s^2$. Although it is hard to conclude the boundedness of $\mathbb{B}^{a}$ from Fig.5.14 (e), the definition of $\mathbb{B}^{tv}$ in (5.16) implies that $\mathbb{B}^{a}$ is also bounded. Fig.5.14 (f) demonstrates the estimation of $\hat{\Theta}$ that includes all unknown parameters and upper bounds. As expected, since the trajectories are time-varying, the state-dependent estimation parameter $\hat{\Theta}$ does not converge to a constant but is under a slow change.

Fig. 5.15 shows the experimental results in comparison with other control approaches, including the fuzzy sliding mode control (Fuzzy-SMC) method [106, p.273-276] and the Radial-Basis-Function-Neural-Network-based adaptive sliding mode control (RBFNN-ASMC) method [106, p.300-303]. In the steady-state phase, both the SMC-Fuzzy approach and the proposed ANTSM approach enable the good position tracking (see Fig. 5.15(a)). However, the Fuzzy-SMC approach fails to provide a smooth convergence to the desired trajectory during the transient phase, as shown in Fig. 5.15(b). As for the RBFNN-ASMC approach, the end effector positions fail to follow the reference signals. We manually stopped the program at $t = 2$ *sec* as the reason for the tracking failure is due to the torque limit of the haptic device. Accordingly, to protect the device, a saturation function is applied to cut off the over-limit generated control inputs. Using the RBFNN-ASMC approach, the plant constantly receives upper/lower limits of the control inputs, instead of the proper control effort that ensures the position tracking. In contrast, using the proposed ANTSM with a logistic time-varying gain allows smooth convergence, and, meanwhile, ensures a high steady-state tracking accuracy.

5.5 Concluding Remarks

A robust ANTSM controller has been presented for a networked leader-following manipulation system. The mixed-type feedback signals and the compensatory bounds are used in the ANTSM controller. The controller ensures the tracking synchronization of the end effectors where the model dynamics are subjected to parametric uncertainties, unknown frictions, unknown external disturbances, and weak topology with time-varying network delays.

With a constant ANTSM control gain, the controller fails to address the hardware limitations, wherea smooth convergence during the transient phase and the high tracking accuracy in the steady-state phase cannot be simultaneously achieved. Therefore, a novel time-varying switching gain has been designed to successfully provide a smooth convergence, and, meanwhile, improves the tracking accuracy during the steady-state phase. The scheme has been successfully applied to a pair of Phantom Omni haptic devices, and relevant parameters to design the time-varying switching gain have been empirically discussed. Rules to design the time-varying switching gain have been generalized as well.

Through numerical simulations and experiments, case studies have been carried out to demonstrate the good position tracking synchronization with bounded errors using the proposed ANTSM control approaches.

Chapter 6

Bilateral Teleoperation Control Design

Another primary subsystem considered in this work is the bilateral telemanipulation system. The bilateral teleoperation robots include one master manipulator that is a human-interface device for taking human commands, and one slave manipulator that interacts with the remote environment. In addition, wireless internet protocol (IP) networks are used to transmit data such as motion signals, force signals, and control inputs. However, due to the long distance and radio frequency interference (RFI), wireless networks can be affected by significant time delays and packet loss, which post challenges to the performance of telemanipulation systems. In this chapter, the development of the ANTSM control approach will be extended to the bilateral teleoperation system[1].

6.1 Problem Descriptions

For the master and slave manipulators, their end effectors are modelled by the nonlinear E-L formulations as

$$\bar{M}_m\ddot{\boldsymbol{x}}_m + \bar{C}_m\dot{\boldsymbol{x}}_m + \bar{\boldsymbol{g}}_m = \boldsymbol{u}_m + \boldsymbol{\rho}_m + \boldsymbol{f}_h, \tag{6.1}$$

$$\bar{M}_s\ddot{\boldsymbol{x}}_s + \bar{C}_s\dot{\boldsymbol{x}}_s + \bar{\boldsymbol{g}}_s = \boldsymbol{u}_s + \boldsymbol{\rho}_s - \boldsymbol{f}_e, \tag{6.2}$$

where the subscripts m and s denote the variables of the master and slave, respectively. In the rest of this chapter, the subscript of $(\cdot)_{m,s}$ will be used to denote the same variables for both the master and slave manipulators. $\boldsymbol{f}_h \in \mathbb{R}^{3\times1}$ defines the human/operator force and $\boldsymbol{f}_e \in \mathbb{R}^{3\times1}$ represents the environmental force. Definitions of the other variables are the same as in (3.28).

---

[1] This chapter was published as the work in [79], 1083-4435 ©2019 IEEE. Reprinted, with permission, from Henghua Shen and Ya-Jun Pan, "Improving Tracking Performance of Nonlinear Uncertain Bilateral Teleoperation Systems with Time-Varying Delays and Disturbances", IEEE/ASME Transactions on Mechatronics, June/2020.

The environment is usually a nonlinear system with unknown dynamics. Nevertheless, the interaction dynamics between the slave arm and environment can be modelled as a spring-damp system. Then, the environmental force can be expressed in a general form [86] as

$$\boldsymbol{f}_e = B_e \dot{\boldsymbol{x}}_s + K_e \boldsymbol{x}_s + C_e \overrightarrow{\mathbf{1}}, \tag{6.3}$$

where $B_e \in \mathbb{R}^{3\times3}$, $K_e \in \mathbb{R}^{3\times3}$, and $C_e \in \mathbb{R}^{3\times3}$ are damping, stiffness and constant respectively. $\overrightarrow{\mathbf{1}} \in \mathbb{R}^{3\times1}$ is a vector of all ones.

Since B_e, K_e, and C_e are unknown parameters, by stacking the unknown parameters into a matrix Θ_e, that is, $\Theta_e = [B_e\ K_e\ C_e] \in \mathbb{R}^{3\times9}$, the environmental force can be rewritten in the following form:

$$\boldsymbol{f}_e = \Theta_e \Phi_e, \tag{6.4}$$

where $\Phi_e = [\dot{\boldsymbol{x}}_s^T\ \boldsymbol{x}_s^T\ \overrightarrow{\mathbf{1}}^T]^T \in \mathbb{R}^{9\times1}$ denotes the known regressor.

The modelling assumptions of the robot and network in this chapter follow those in (3.28) and (3.31), respectively. In addition, We assume that both the human operator force $\boldsymbol{f}_h$ and the environmental force $\boldsymbol{f}_e$ are measurable,

Control Objective

The objective is to drive the states of slave manipulators towards the state of the master that is commanded by a human operator such that $\|\boldsymbol{x}_m(t) - \boldsymbol{x}_s(t)\|_1 \leq \epsilon$ with a small bound $\epsilon \geq 0$. Controllers to be designed should be capable of dealing with the unknown modelling information (including parametric uncertainties, unknown frictions, and external disturbances) and unknown time-varying delays. In addition, to avoid direct force transmission, the master controller is required to predict the real-time environmental force feedback with a small force estimation error.

6.2 Control Development

The bilateral telemanipulation system that will be considered is illustrated in Fig.6.1. A force estimator and a force predictor allow the environmental force to be recursively predicted in the master side without directly transmitting force signals over the network channels. Reference signals are generated from the human input and

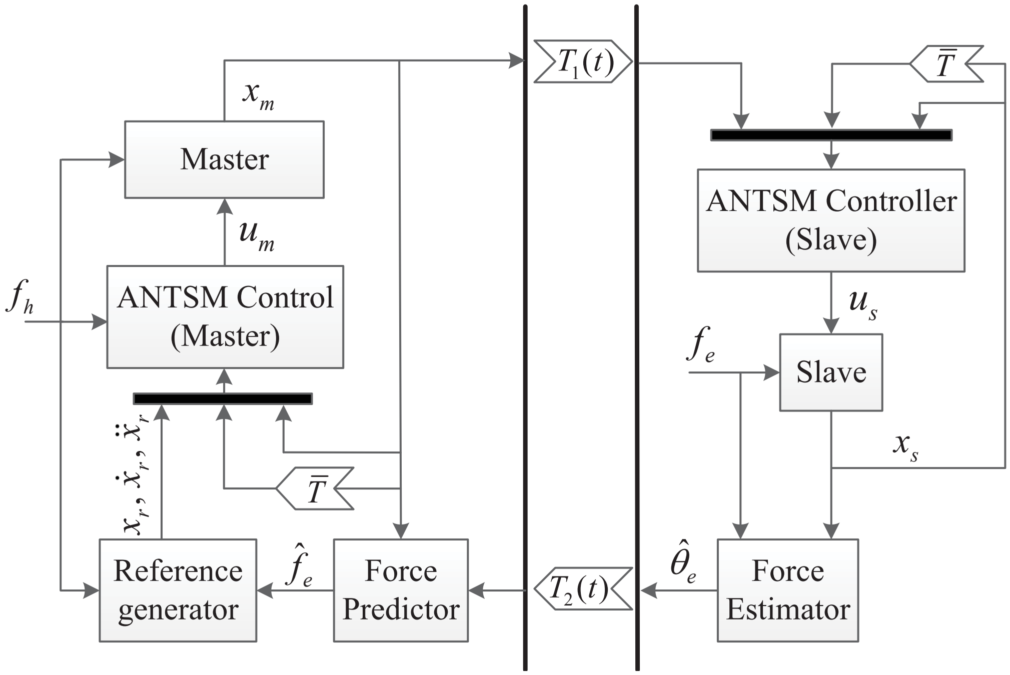


Figure 6.1: Bilateral teleoperation control architecture.

the predicted environmental force through a reference signal generator. As a result, tasks for the ANTSM controllers (in both the master side and the slave side) can be merely simplified to motion control. In what follows, the force prediction and ANTSM control methods will be introduced.

6.2.1 Environmental Force Prediction

Since forces are power signals and their direct transmission over the network channels may cause instability problems. One of the solutions is to transform the environmental force into non-power signals that are actually transmitted. Consider the linearization in (6.4), the environmental force information can be carried by an unknown force parameter $\hat{\Theta}_e$ [107] which is estimated by

$$\dot{\hat{\Theta}}_e = -H\Phi_e(\Phi_e^T\hat{\Theta}_e - \boldsymbol{f}_e), \tag{6.5}$$

$$\frac{d}{dt}H^{-1} = \Phi_e\Phi_e^T, \quad H(0) = h_0 I_{3\times 3}, \quad h_0 > 0. \tag{6.6}$$

Then, instead of the force signal itself, the non-power parameter $\hat{\Theta}_e$ is sent to the master side. If the regressor Φ_e is also fed back to the master side, the environmental force can be recovered easily by

$$\hat{\boldsymbol{f}}_e(t - T_2(t)) = \hat{\Theta}_e(t - T_2(t))\Phi_e(t - T_2(t)). \tag{6.7}$$

As can be seen from (6.7), the recovered force is delayed, and the human operator would feel large force sluggish if the network experiences large latencies. Therefore, assume that the controllers provide finite-time motion synchronization between the master and slave, the regressor in the force predictor can use the current master states, instead of the network-delayed slave states, to reduce the effect of the network delays. Let $\Phi_m = [\dot{\boldsymbol{x}}_m^T, \boldsymbol{x}_m^T, \overrightarrow{\mathbf{1}}^T]^T$, the environmental force can be predicted by

$$\hat{\boldsymbol{f}}_e = \hat{\Theta}_e(t - T_2(t))\Phi_m. \tag{6.8}$$

6.2.2 Master Controller Design

To allow the master controller to be designed for the purpose of motion tracking only, the human force input and the estimated environmental force generate reference signals through a target impedance model given by

$$M_r\ddot{\boldsymbol{x}}_r + C_r\dot{\boldsymbol{x}}_r + \boldsymbol{g}_r = k_{sc1}\boldsymbol{f}_h - k_{sc2}\hat{\boldsymbol{f}}_e, \tag{6.9}$$

where $\boldsymbol{x}_r \in \mathbb{R}^{3\times1}$, $\dot{\boldsymbol{x}}_r \in \mathbb{R}^{3\times1}$, and $\ddot{\boldsymbol{x}}_r \in \mathbb{R}^{3\times1}$ denote the reference position, velocity, and acceleration for the motion control design. k_{sc1} and k_{sc2} are the scaling factors for the human operator force and the estimated environmental force, respectively. $M_r \in \mathbb{R}^{3\times3}$, $C_r \in \mathbb{R}^{3\times3}$, $\boldsymbol{g}_r \in \mathbb{R}^{3\times1}$ are the target parameters of the master behavior. With the reference signals from (6.9), feedback signals for the subsequent master controller design are given by

$$\begin{cases} \boldsymbol{e}_m^p = \boldsymbol{x}_m - \boldsymbol{x}_r, \\ \boldsymbol{e}_m^v = \dot{\boldsymbol{x}}_m - \dot{\boldsymbol{x}}_r, \end{cases} \tag{6.10}$$

where $\boldsymbol{e}_m^p \in \mathbb{R}^{3\times1}$ is the position error and the velocity error is denoted by $\boldsymbol{e}_m^v \in \mathbb{R}^{3\times1}$ for the subsequent design. From (3.36), the sliding surface $\boldsymbol{s}_m \in \mathbb{R}^{3\times1}$ is given by

$$\boldsymbol{s}_m = \boldsymbol{e}_m^p + \beta_m(\boldsymbol{e}_m^v)^{\alpha_m}. \tag{6.11}$$

With the sliding surface in (6.11), the The adaptive controller for the developed master manipulator is described in the following theorem.

Theorem 6.1. *Let the adaptive law be given by*

$$\dot{\hat{\Theta}}_m = -\alpha_m\beta_m\|\boldsymbol{s}_m^T diag[(\boldsymbol{e}_m^v)^{(\alpha_m-1)}]\|_1\bar{M}_m^{M+}\Phi_m,$$

the ANTSM controller for the master manipulator is designed by

$$\begin{aligned}\boldsymbol{u}_m &= \bar{C}_m\dot{\boldsymbol{x}}_m + \bar{\boldsymbol{g}}_m - \boldsymbol{f}_h + \bar{M}_m\Big\{\frac{-(\boldsymbol{e}_m^v)^{(2-\alpha_m)}}{\alpha_m\beta_m} + \ddot{\boldsymbol{x}}_r \\ &\quad -\bar{M}_m^{M+}\hat{\Theta}_m\Phi_m sgn(\boldsymbol{s}_m) - \kappa_m sgn(\boldsymbol{s}_m)\Big\}. \qquad (6.12)\end{aligned}$$

where $\kappa_m > 0$, $\bar{M}_m^{M+} \geq \|\bar{M}_m^{+}\|_1$ *and* $\hat{\Theta}_m$ *is the estimate of* Θ_m.

See the stability proof in Appendix C.1.

6.2.3 Follower Controller Design

Unlike the master controller, the slave's reference signals (i.e., master states) are delayed and, therefore, a phase shift is inevitable. Consider the functionalities of different types of feedback signals in Section 4.2.1 as well as the benefits of using the mixed-type feedback in (5.3) and (5.4), the mixed-type feedback to compute the slave controller is given by

$$\begin{cases} \boldsymbol{e}_s^p = \boldsymbol{x}_s(t-\bar{T}) - \boldsymbol{x}_m(t-T_1(t)), \\ \boldsymbol{e}_s^v = \dot{\boldsymbol{x}}_s(t) - \dot{\boldsymbol{x}}_m(t-T_1(t)), \end{cases} \qquad (6.13)$$

Then, the relationship between the position and velocity errors can be established by an auxiliary self-delayed velocity error, $\boldsymbol{\varepsilon}_s^{sd} \in \mathbb{R}^{3\times1}$, that is,

$$\begin{aligned}\boldsymbol{\varepsilon}_s^{sd} &= \boldsymbol{e}_s^v - \dot{\boldsymbol{e}}_s^p \\ &= \dot{\mathbf{x}}_s(t) - \dot{\mathbf{x}}_s(t-\bar{T}) - \dot{\mathbf{x}}_m(t-T_1(t))(1+\dot{\delta}_1(t)). \qquad (6.14)\end{aligned}$$

The self-delay-induced error is upper-bounded as follows.

$$\mathbb{B}_s^{sd} = \|\dot{\mathbf{x}}_s(t) - \dot{\mathbf{x}}_s(t-\bar{T})\|_1 + 2\|\dot{\mathbf{x}}_m(t-T_1(t))\|_1, \qquad (6.15)$$

such that $\|\boldsymbol{\varepsilon}_s^{sd}\|_1 \leq \mathbb{B}_s^{sd}$ holds. To deal with another error caused by time-varying delays when the acceleration signals are acquired by time derivative, a compensatory bound $\mathbb{B}_s^a$ is introduced as follows.

$$\mathbb{B}_s^a(t) = 2||\ddot{x}_m(t-T_1(t))||_1, \qquad (6.16)$$

The sliding surfaces of $\boldsymbol{s}_s \in \mathbb{R}^{3\times 1}$ for the control design are given by

$$\boldsymbol{s}_s = \boldsymbol{e}_s^p + \beta_s(\boldsymbol{e}_s^v)^{\alpha_s}, \tag{6.17}$$

and the adaptive controllers are designed as

$$\begin{aligned} \boldsymbol{u}_s &= \bar{C}_s\dot{\boldsymbol{x}}_s + \bar{\boldsymbol{g}}_s + \bar{M}_s\Big\{\frac{-(\boldsymbol{e}_s^v)^{(2-\alpha_s)}}{\alpha_s\beta_s} + \ddot{\boldsymbol{x}}_m(t - T_1(t)) \\ &\quad -\Big(\frac{1}{\alpha_s\beta_s}\mathbb{B}_s^{sd}\mathrm{diag}(\boldsymbol{e}_s^v)^{(1-\alpha_s)} + \mathbb{B}_s^a\Big)\mathrm{sgn}(\boldsymbol{s}_s) \\ &\quad -\bar{M}_s^{M+}\hat{\Theta}_s\Phi_s\mathrm{sgn}(\boldsymbol{s}_s) - \kappa_s\mathrm{sgn}(\boldsymbol{s}_s)\Big\} + \boldsymbol{f}_e, \end{aligned} \tag{6.18}$$

where $\hat{\Theta}_s$ is the estimate of Θ_s. The following theorem presents the adaptive law to design $\hat{\Theta}_s$, which guarantees the finite-time stability of the slave control system.

Theorem 6.2. *For the slave manipulator in a teleoperation system, the ANTSM controller in (6.18) with the compensatory bounds updated by (6.15) and (6.16), and the adaptive law given by*

$$\dot{\hat{\Theta}}_s = -\alpha_s\beta_s\|\boldsymbol{s}_s^T \mathit{diag}[(\boldsymbol{e}_s^v)^{(\alpha_s-1)}]\|_1\bar{M}_s^{M+}\Phi_s, \tag{6.19}$$

guarantees the system stability and the sliding surface $\boldsymbol{s}_s = \boldsymbol{0}$ in (6.17) to be reached in a finite time t_s that satisfies

$$t_s \le \frac{\sqrt{6}V_s(x_s(0))^{\frac{1}{2}}}{3\alpha_s\beta_s\kappa_s\|(\boldsymbol{e}_s^v)^{(\alpha_s-1)}\|_1}. \tag{6.20}$$

See the proof in Appendix C.2.

As using the mixed-type feedback does not always provide optimal performance, the following theorem provides sufficient condition that demonstrates the outperformance of the mixed-type feedback.

Theorem 6.3. *When $\boldsymbol{s}_s = \boldsymbol{0}$ is satisfied, the ANTSM controller in (6.18) guarantees the slave manipulator to track the position of the master manipulator with bounded tracking errors. In addition, the tracking error bound indicates that using the mixed-type feedback signal outperforms the utilization of feedback without self-delays by selecting the control gains that satisfy the following sufficient condition:*

$$(4^{\alpha_s}\beta_s)^{\frac{1}{1-\alpha_s}} < \mathbb{B}_v\bar{T}^{\frac{1}{1-\alpha_s}}, \tag{6.21}$$

where $\mathbb{B}_v$ denotes the master velocity bound.

See the proof in Appendix C.3.

From the analysis of Theorem 6.2 and Theorem 6.1, two facilitating propositions can be concluded as follows.

Proposition 6.1. *Utilizing the mixed-type feedback signal in the slave control design improves the velocity tracking between the master and the slave in the sense that*

$$\|\dot{\boldsymbol{x}}_s(t-\bar{T})\|_1 \rightarrow \|\dot{\boldsymbol{x}}_m(t-T_1(t))\|_1,$$

which explains the reduction of the delay-induced phase shift in the position profiles between the master and slave.

See the proof in Appendix C.4.

Proposition 6.2. *After* $\boldsymbol{s}_s = \boldsymbol{0}$ *is satisfied, provided that the reference signal is constant with zero speed, the tracking errors* $\boldsymbol{e}^p_{m,s}$ *tend to converge to zero, that is,* $\boldsymbol{e}^p_{m,s}(t) \rightarrow \boldsymbol{0}$ *as* $t \rightarrow t_s$*.*

See the proof in Appendix C.5.

6.3 Numerical Simulations

A numerical study was carried out using mathematical models of the 3-DOF ($k = 3$) Phantom Omni haptic devices, as described in Section 3.3.3.

Unless otherwise specified, the initial setting for the manipulators is as follows. The manipulator joints are initially static and their positions are initialized as (unit: rad): $\boldsymbol{q}_s(0) = [0.700\ 0.600\ -0.500]^T$, $\boldsymbol{q}_m(0) = [-0.300\ 0.300\ -0.800]^T$.

A virtual human force exerted on the master manipulator is given by: $[0.030\sin(ft) + 0.100\ 0.070\ 0.040\sin(ft)]^T\ N$ with $f = pi/10$. The virtual environmental force in the slave side is assumed to be static as $[0.025\ 0.040\ 0.030]^T\ N$. The assigned velocity is upper-bounded by $\mathbb{B}_v = 0.250\ m/s$. The random time-varying network delay is given in a bound as $T_1(t) = T_2(t) = 0.3 \pm 0.002\ s$. To satisfy the condition (6.21) in Theorem 6.3, the control gains are chosen as $\alpha_{m,s} = 7/5$, $\beta_{m,s} = 1$, and $\kappa_{m,s} = 8$. The frictional force coefficients are $\gamma^1_{m,s} = 3, \gamma^2_{m,s} = 6, \gamma^3_{m,s} = 1, \gamma^4_{m,s} = 2, \gamma^5_{m,s} = 4, \gamma^6_{m,s} = 1$. Assume that the parameter uncertainties are half of the true values, that is, $\Delta M_{m,s} = \frac{1}{2}M_{m,s}$, $\Delta C_{m,s} = \frac{1}{2}C_{m,s}$, and $\Delta\boldsymbol{g}_{m,s} = \frac{1}{2}\boldsymbol{g}_{m,s}$. $M_r = diag[2\ 2\ 1.4]$, $C_r = diag[1.5\ 2\ 1.5]$,

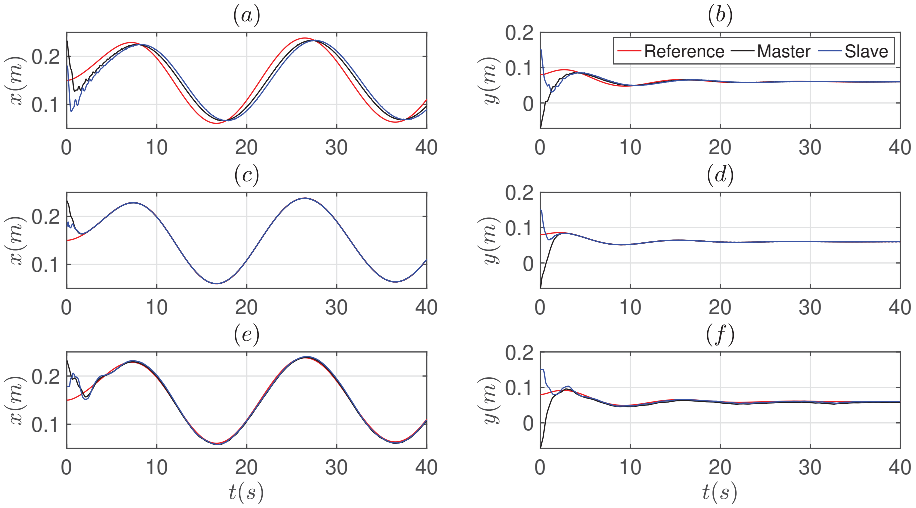


Figure 6.2: The end effector position tracking synchronization of the master and the slave (along x and y directions) with controllers using different types of feedback as: Case I: (a) and (b) using feedback without self-delays ($T = 0.3\ sec$); Case II: (c) and (d) using feedback without self-delays ($T = 0.02\ sec$); Case III: (e) and (f) using the mixed-type feedback ($T = 0.3\ sec$).

$\boldsymbol{g}_r = [2.5\ 2\ 1.6]^T$, and $k_{sc1} = k_{sc2} = 1$ as in (6.9). As discussed previously, in the sliding mode controllers, a saturation function replaces the sign function to reduce the chattering effect.

6.3.1 Comparison of Different Types of Feedbacks

As discussed previously, it is beneficial to use the mixed-type feedback signals in the ANTSM controllers. By purposely assigning time-varying reference signals and constant reference signals along x and y directions, respectively, results in Fig.6.2 demonstrate the effects of different types of feedback signals as:

Case I: Fig.6.2 (a) and (b) show the results of using feedback without self-delays with $\bar{T} = 0.3\ s$;

Case II: Fig.6.2 (c) and (d) show the results of using feedback without self-delays with $\bar{T} = 0.02\ s$. Notice that a smaller nominal delay interval $\bar{T} = 0.02\ s$ was chosen in this case because the controllers failed to maintain the stability when $\bar{T} > 0.02\ s$.

Case III: Fig.6.2 (e) and (f) show the results of using the mixed-type feedback with $\bar{T} = 0.3\ s$.

The results in Fig.6.2 (a), (c), and (e) are consistent with those in Fig. 4.2 that

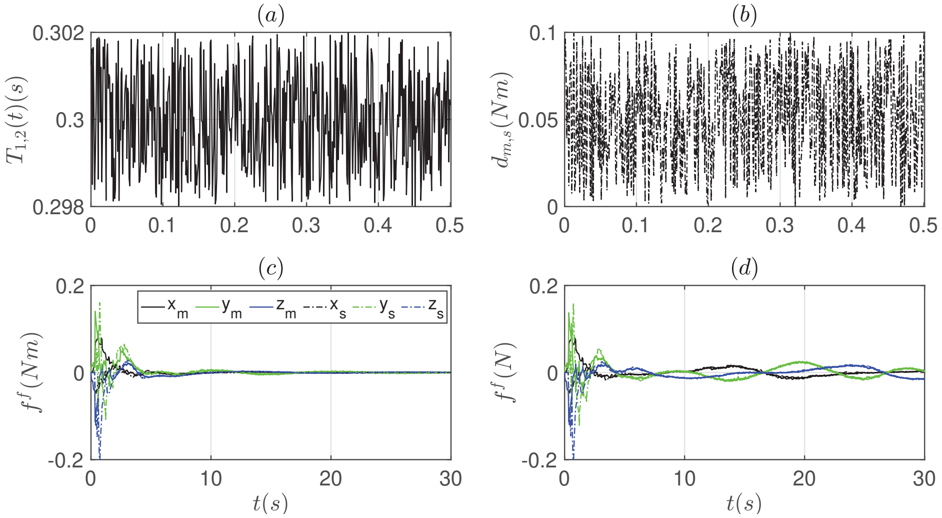


Figure 6.3: Plots of the (a) Network delays; (b) External disturbances and the frictional forces in (c) Case 1 and (d) Case 2.

using the mixed-type feedback significantly improves the tracking accuracy, although the network delay is long. However, Fig.6.2 (b), (d), and (f) demonstrate that, given constant reference signal, there is no much difference using whichever type of feedback signals. This agrees with Proposition 6.2 and explains why the wait-and-move operations are preferred in many time-delayed applications to avoid the phase-shift effect.

6.3.2 Tracking Performance Illustration

Performance of the bilateral telemanipulation system is demonstrated with the following two cases:

Case 1: The motion commands sent from the master manipulator are constant signals.

Case 2: The motion commands sent from the master manipulator are time-varying signals.

The random network delays and external disturbance are shown in Fig.6.3 (a) and (b), respectively. Fig.6.3 (c) and (d) present the frictional forces in **Case 1** and **Case 2**. As the friction force is joint-velocity dependent, in **Case 1**, velocities of the slave joints approximate to zero when the joint positions converge to the constant reference signals. As a result, frictional forces go to zero in **Case 1**. In contrast, the non-zero

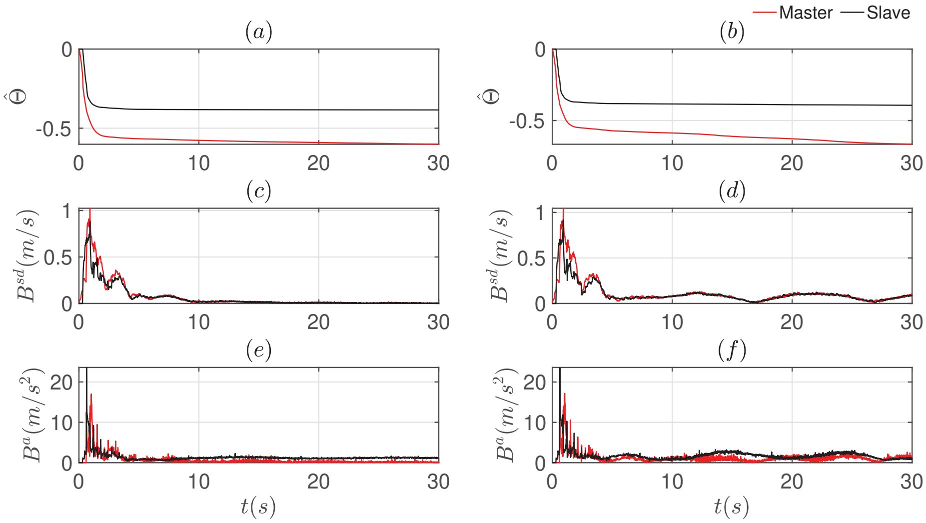


Figure 6.4: (a) Estimation parameters $\hat{\Theta}$ in Case 1; (b) Estimation parameters in Case 2; Compensatory bound $\mathbb{B}_s^{sd}$ (c) in Case 1 and (d) in Case 2; Compensatory bound $\mathbb{B}_s^{a}$ (e) in Case 1 and (f) in Case 2.

frictions exist throughout the operation in **Case 2**.

Similarly, the simulation results in Fig.6.4 shows that estimations of the unknown parameters Θ and the compensatory bounds are continuously time-varying in **Case 2**. Nevertheless, the plots of the unknown parameter estimations and compensatory bounds are all bounded and varying slowly in **Case 2**, which reveals the good adaptive ability of the controller in dealing with the unknown dynamics models. In addition, in Fig.6.4 (a), (c), and (e) the velocity-dependent variables are, as expected, close to zero during the steady-state phase in **Case 1**.

Fig.6.5 and Fig.6.6 demonstrate that asymptotic synchronization of the master and slave manipulators for the developed controller, despite the inexact information of system parameters, external noise bounds, time-varying network delays, and frictional forces. The tracking errors are defined by the difference between the reference trajectories and those of the master and the slave, respectively. In both cases, the tracking errors appear to be within small bounds. For example, the error bounds are $[-3.179,\ -0.105] \times 10^{-3}\ m$ in **Case 1**, and $[-3.684,\ 4.167] \times 10^{-3}\ m$ in **Case 2**. In addition, synchronizations toward the reference signals are all in a finite time, i.e., about 5 s.

The performance of the environmental force predictor is illustrated in Fig.6.7,

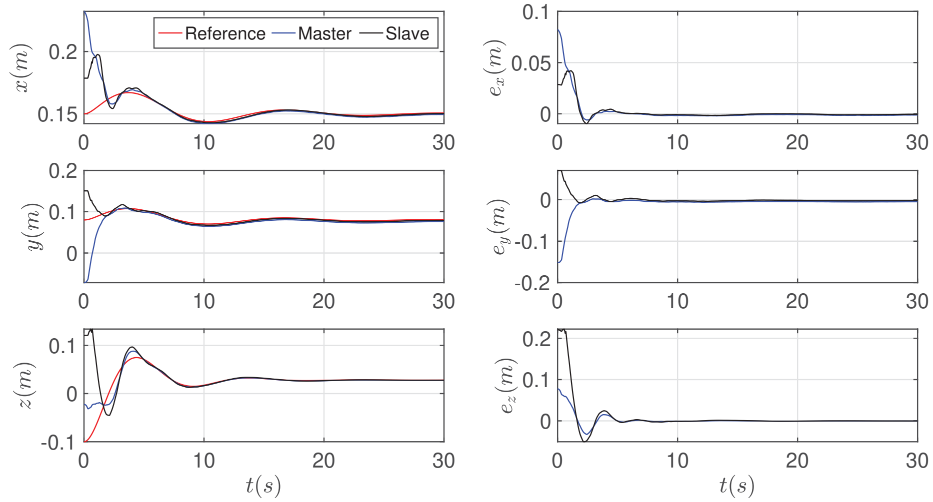


Figure 6.5: The position tracking and tracking errors with constant reference signals as in **Case 1**.

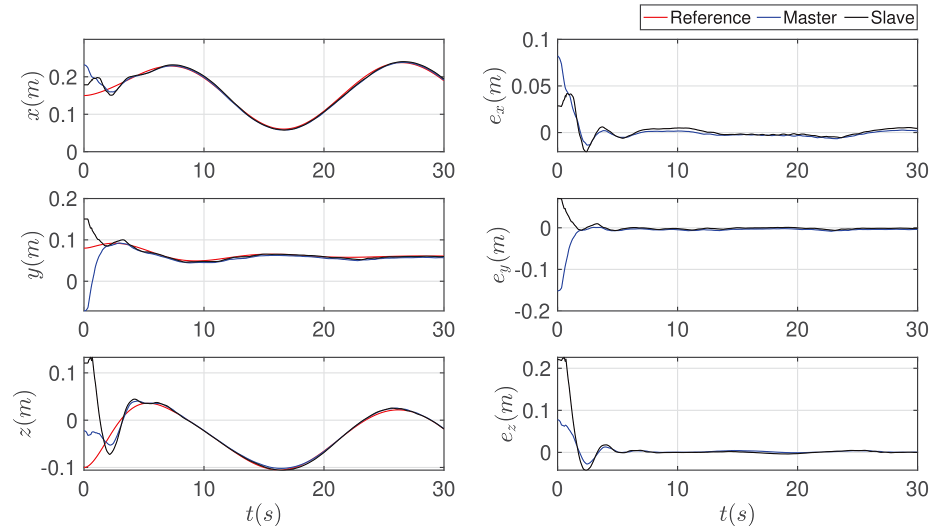


Figure 6.6: The position tracking and tracking errors with time-varying reference signals as in **Case 2**.

which shows that the 3D environmental force can be accurately predicted in the master side in both **Case 1** and **Case 2**.

6.3.3 Comparisons with existing control methods

Some additional simulation results are provided which demonstrate the enhanced performance of the proposed controller compared to other prominent controllers, namely, a model-free P-like control method in [88] and a model-based NTSM control method

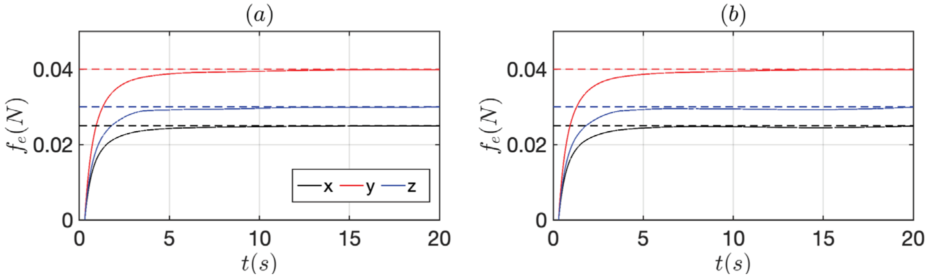


Figure 6.7: Environmental force estimations for (a) **Case 1** and (b) **Case 2** where the solid lines represent the true environment forces measured in the slave side and and dash lines represent the according predicted forces in the master side.

Table 6.1: Comparisons of three control methods

Control Method	Features			
	Self-Delay	Adaptiveness	Robustness	Force Transmission
P-like [88]	No	No	No	Wave Form
NTSM [108]	No	No	Yes	Direct Transmission
ANTSM	Yes	Yes	Yes	Force Prediction

Table 6.2: Steady-state errors using three different controllers

	Errors ($\times 10^{-3}\ m$)	Control Methods		
		P-like [88]	NTSM [108]	ANTSM
Constant reference	Mean	1.933	-2.822	-1.284
	SD	0.522	1.839	1.143
	Max	2.869	0.078	-0.105
	Min	0.962	-4.808	-3.179
Time-varying reference	Mean	-0.947	-2.615	-0.952
	SD	33.933	4.400	1.959
	Max	74.104	6.643	4.167
	Min	-66.246	-10.613	-3.684

SD: Standard Deviation.

in [108].

The P-like control method is based on the passivity theory and does not require to know the dynamic model. Therefore, the P-like controller is able to handle the dynamic uncertainties. The NTSM method, however, is model-based. An NTSM controller provides high tracking accuracy when the dynamic parameters are exactly

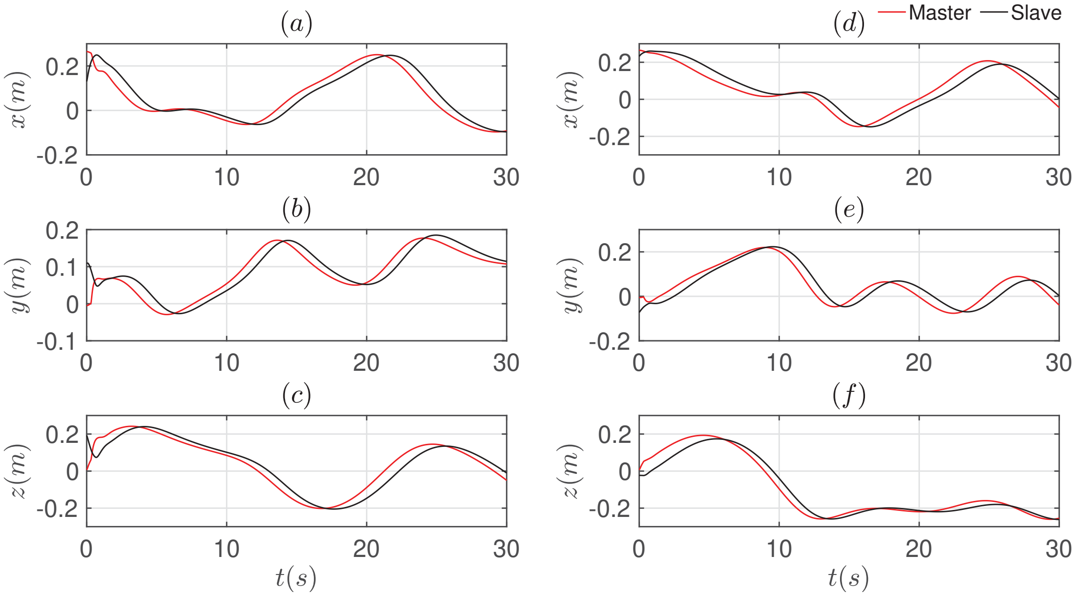


Figure 6.8: The master-slave position tracking in Cartesian space with time-varying reference signals using P-like controller in [88]. Left column: without noises; right column: with noises. (Note that the trajectories are different from those in Fig.6.6 because the controllers are designed in joint space.)

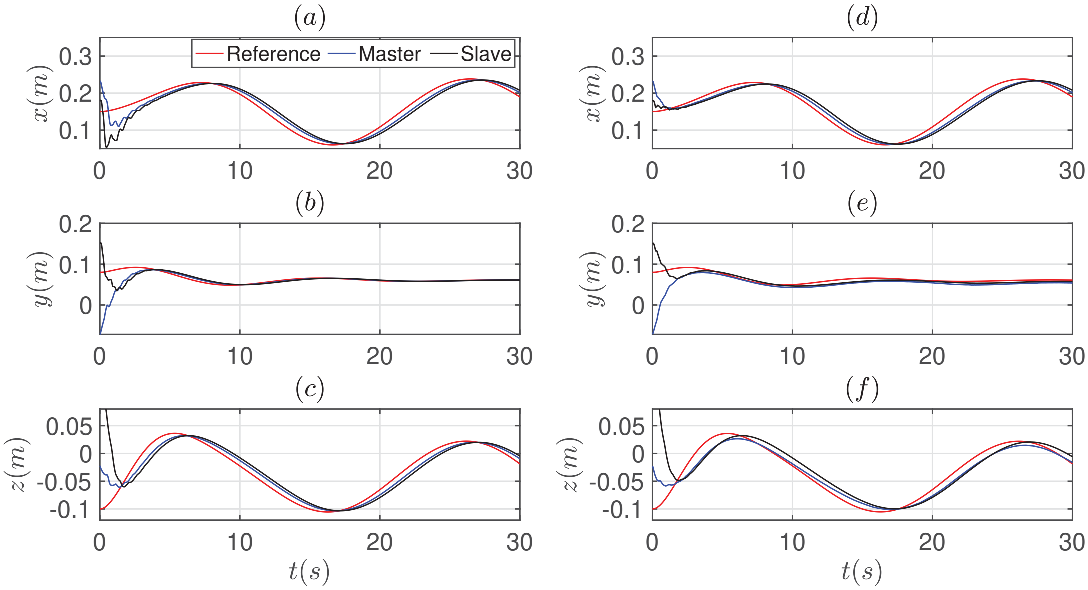


Figure 6.9: The master-slave position tracking in Cartesian space with time-varying reference signals using a model-based NTSM controller in [108], using the feedback signals without self-delays. Left column: without noises; right column: with noises.

known. In addition, like other sliding mode control methods, it facilitates the robust control once the disturbance bounds are known. The major features of the compared control methods are listed in Table.6.1, including the utilization of feedback types, adaptive ability, robust ability, and methods of the environmental force transmission.

The tracking error comparisons of the three controllers are presented in Table.6.2. Given constant reference signals, the three control schemes provide the similar tracking performance with small tracking error bounds. However, when the reference signals are time-varying, using P-like controller results in large tracking errors with a standard deviation of $33.933 \times 10^{-3}\ m$). The NTSM method presents considerably smaller average tracking errors. However, a significant variation between the maximum and minimum values can still be observed. In contrast, the proposed ANTSM scheme can provide better tracking accuracy regardless of the time-dependent changes of the reference.

The results in Fig.6.8 and Fig.6.9 are presented to highlight the proposed ANTSM controller's robustness and ability in dealing with the phase shift effect. Fig.6.8 shows that the P-like controller cannot reduce the phase shift caused by network delays. As a result, the delay-induced tracking error persists. In addition, although the operation is executed with the same human force and environmental force, the external disturbances have significantly changed the behaviours of the master and slave manipulators, as can be seen from Fig.6.8 (d)-(f), in comparison with those in Fig.6.8 (a)-(c). Therefore, for systems with unknown external disturbances, the P-like controller may not provide expected motions and operations.

As depicted in Fig.6.9, the NTSM controller allows the systems to maintain desired performance in the presence of bounded external disturbances. However, similar to the P-like controller, the NTSM method cannot address the issue of the delay-induced tracking error. In addition, the NTSM method is model-based, so the tracking errors may further increase when the initial dynamic uncertainties are large.

By comparing the results in Fig.6.8, Fig.6.9 and Fig.6.6, we can see that the proposed ANTSM controller has a good ability in dealing with model uncertainties, external noises, and the delay-induce phase-shift effect.

6.4 Experimental Results

In this section, experiments were conducted on a pair of 3-DoF Phantom Omni haptic devices served as the master and slave, respectively. The experimental setup is

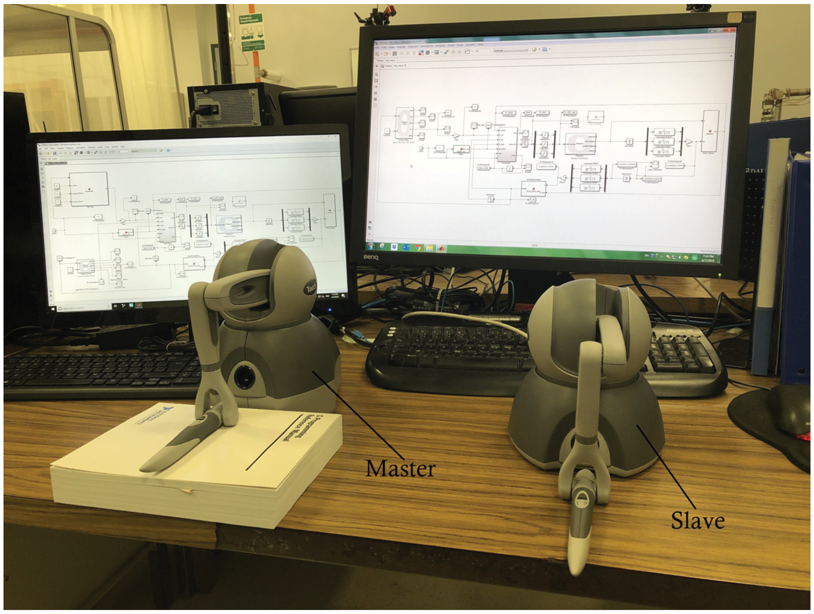


Figure 6.10: The teleoperation experimental testbed with two network-connected Phantom Omni haptic devices.

presented in Fig.6.10. Two devices are run in two different computers connected through a wireless LAN. The one-way delays of the User Datagram Protocol (UDP) send/receive are with an average network delay of $\bar{T} = 0.05\ s$ and a standard error of $\sigma = 0.005\ s$. The control gains were set to: $\alpha_m = \alpha_s = 7/5$, $\beta_m = \beta_s = 1$, and $\kappa_m = \kappa_s = 8$.

Tests with pre-recorded time-varying reference trajectories were carried out and the end effector position tracking is presented in Fig.6.11 (a)-(c). It shows that the ANTSM controller guarantees the good tracking synchronization between the master and the slave, with small and bounded tracking errors. As shown in Fig.6.11 (d)-(f), the tracking errors are bounded within $[-0.005\ 0.005]\ m$.

To protect the devices from exceeding input limits, saturation blocks in the Simulink model were used to constrain the control inputs to $\boldsymbol{\tau}_{m,s} \in [-0.200\ 0.200]\ Nm$. From Fig.6.12, the master and slave controllers are able to generate torque inputs that satisfy the safety requirements. In operating of the slave manipulator, during the steady-state phase, the two compensatory bounds are constrained within a small range, i.e., $\mathbb{B}_s^{sd} \leq 0.004\ m/s$ and $\mathbb{B}_s^{a} \leq 0.150\ m/s^2$, as depicted in Fig.6.13 (a) and (b), respectively. Fig.6.14 (a) depicts good ability of the force predictor to estimate the virtual environment forces. Fig.6.14 (b) demonstrates the estimation parameters

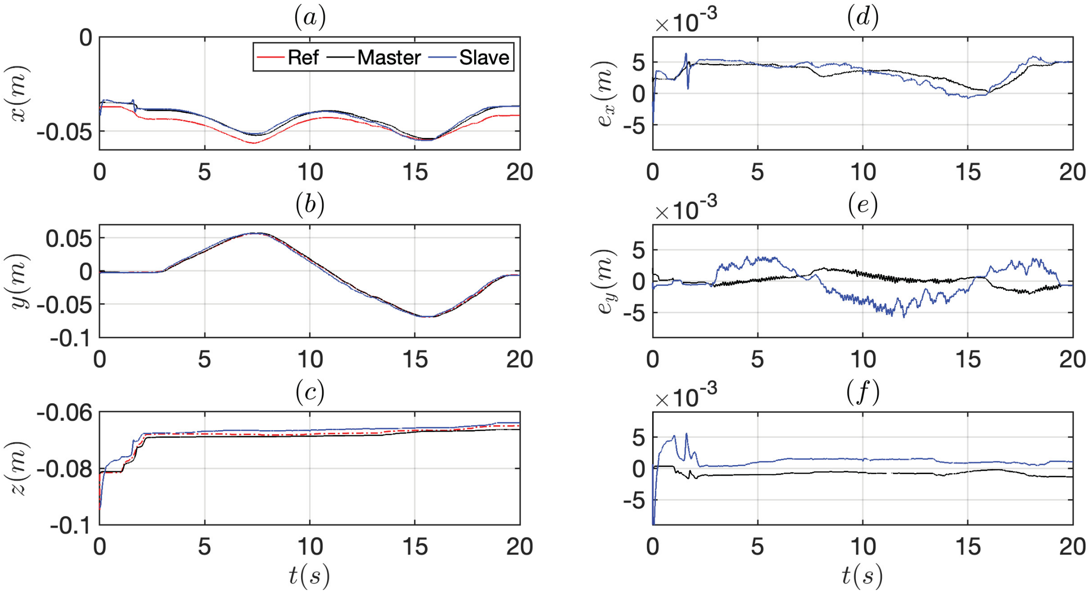


Figure 6.11: Position tracking and the tracking error of the master-slave system with time-varying reference signal.

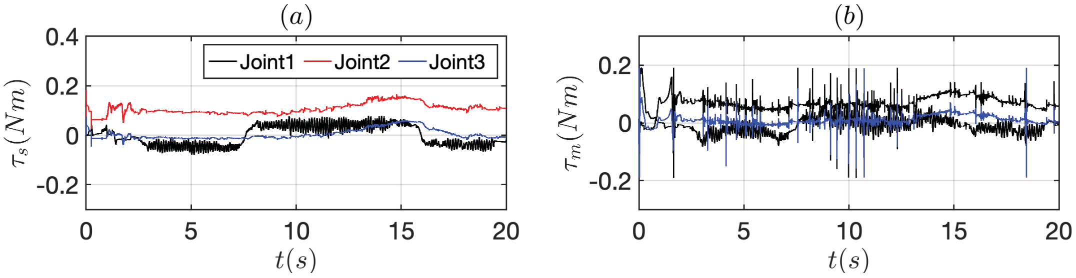


Figure 6.12: Control inputs for joint motors of (a) the master and (b) the slave manipulators.

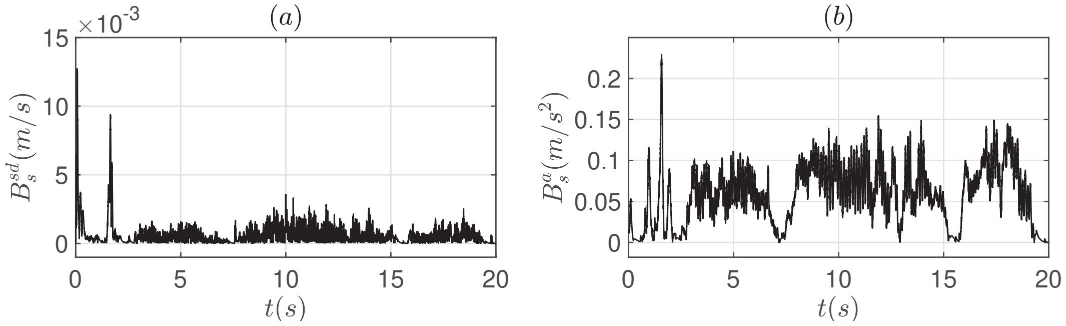


Figure 6.13: The compensatory bounds of (a) $\mathbb{B}_s^{sd}$ and (b) $\mathbb{B}_s^{a}$ for the slave controller.

for the master and slave systems. As the trajectories are time-varying, the estimation parameters do not converge to a constant.

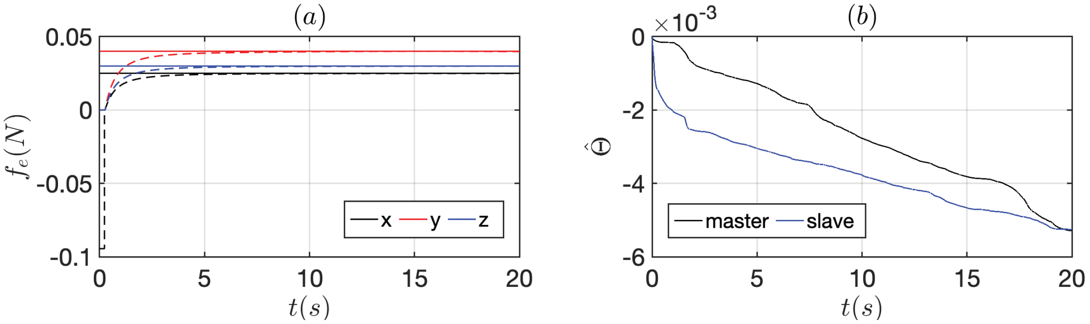


Figure 6.14: Plots of (a) the estimated environmental forces (solid lines) and the true environmental forces (dash lines), and (b) the estimation of the unknown dynamic parameters.

6.5 Concluding Remarks

The ANTSM control approaches were developed for a bilateral teleoperation system in the presence of time-varying network delays and unknown model information. Results of the simulations and experiments carried out in this chapter show that the ANTSM controllers using the mixture of feedback signals, as well as the compensatory bounds, work well in improving the motion tracking performance of the slave manipulator. The use of force estimation and prediction reduces the network-delayed effect on the force transmission, which enables the human operator to perceive the environmental force in a remote environment.

Chapter 7

Teleoperation of Multiple Cooperative Slave Manipulators

This chapter[1] aims at building a single-master-multiple-slave (SMMS) manipulation system based on the techniques developed for multi-manipulator systems and bilateral teleoperation systems in the earlier chapters. Specifically, techniques developed for the SMMS system include the full pose control method as in Chapter 4, adaptive control approaches as Chapter 5, and bilateral teleoperation control and force prediction as in Chapter 6. The distributed control framework facilitates the integration of the two subsystems into one complex system without changing a significant amount of the control architecture for each manipulator. Additionally, the multi-manipulator system in the slave side is constructed in a cooperative manner. Therefore, the motion distribution and load allocation[2] strategies are taken into account. As well, concepts from the ANTSM control approach are used for the pose tracking of the networked SMMS manipulation system in the presence of time-varying delays, model uncertainties, and external disturbances. Simulation results of Phantom Omni haptic devices demonstrate the effectiveness of the proposed control design.

7.1 Problem Description

An SMMS system consists of one master manipulator, one leader manipulator on the slave side, and the follower manipulators that are driven to track the leader manipulator's states. The SMMS system is illustrated in Fig.7.1. Reference motion signals

---

[1] This chapter was partially published as the work in [78], ©2017 IEEE. Reprinted, with permission, from Henghua Shen, Ya-Jun Pan and Bingwei He, "Teleoperation of multiple cooperative slave manipulators using graph-based non-singular terminal sliding-mode control", 2017 IEEE International Conference on Robotics and Biomimetics, December/2017, and partially published as the work in [109], ©2019 IEEE. Reprinted, with permission, from Henghua Shen, Ya-Jun Pan and Georgeta Bauer, "Manipulability-Based Load Allocation and Kinematic Decoupling in Cooperative Manipulations", 2019 IEEE 28th International Symposium on Industrial Electronics, June/2019.

[2] In this chapter, the concept of load, also referred to as wrench, consists of linear force and torque that is the rotational equivalent of linear force. Unless otherwise specified, the force in the remaining chapter refers to the linear force.

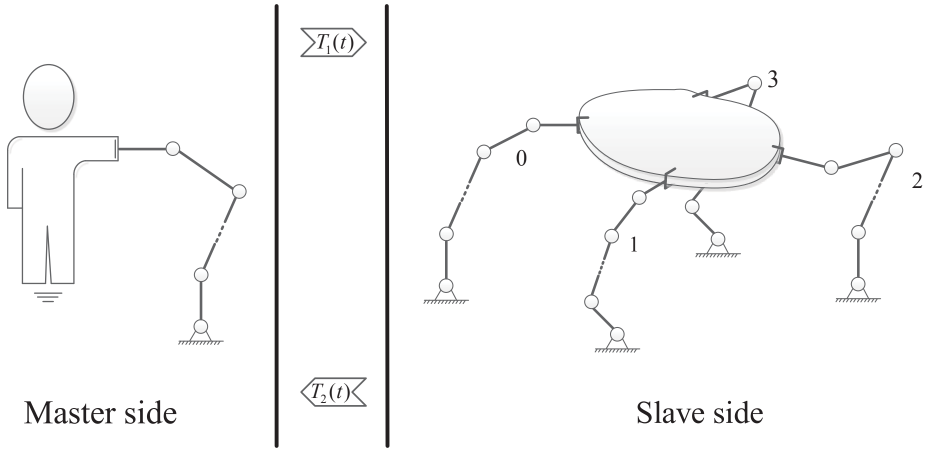


Figure 7.1: Control architecture of an SMMS manipulation system.

for the slave manipulators are given by the master robot arm manipulated by the human operator. On the slave side, manipulators form a leader-following cooperative framework with a common task, e.g., handling a common object. Therefore, not all of the slave robots are required to communicate with the master agent. Instead, the master only mutually communicates with the leader, while the rest of the slave manipulators obtain the master information through the leader. In the remaining text, the slaves, except the leader, are referred to as followers. The robot dynamics are modelled as follows.

$$\bar{M}_m\ddot{\boldsymbol{x}}_m + \bar{C}_m\dot{\boldsymbol{x}}_m + \bar{\boldsymbol{g}}_m = \boldsymbol{u}_m + \boldsymbol{\rho}_m + \boldsymbol{h}_h, \tag{7.1}$$

$$\bar{M}_i\ddot{\boldsymbol{x}}_i + \bar{C}_i\dot{\boldsymbol{x}}_i + \bar{\boldsymbol{g}}_i = \boldsymbol{u}_i + \boldsymbol{\rho}_i - \boldsymbol{h}_i, \tag{7.2}$$

where $(*)_m$ denotes variables of the master manipulator and $(*)_i$ denotes the slave variables with $i = 0$ for the leader and $i = 1, 2, \cdots, n$ $(n \geq 1)$ for the followers. Detailed definitions of the variables can be found in (3.28). As we consider the full pose operation, dimensions of the variables are as follows. $\boldsymbol{x}_m,\ \boldsymbol{x}_i\ \in \mathbb{R}^7$ denote the poses that consist of position and orientation. For example,

$$\boldsymbol{x}_m = [\boldsymbol{p}_m^{\mathrm{T}} \quad \boldsymbol{\xi}_m^{\mathrm{T}}]^{\mathrm{T}},$$

where $\boldsymbol{p}_m \in \mathbb{R}^3$ is the translational position and the unit quaternion $\boldsymbol{\xi}_m = [\eta_m \ \ \boldsymbol{\epsilon}_m^{\mathrm{T}}]^{\mathrm{T}} \in S^3$ represents the orientation. η_m is a scalar representing the real part of the quaternion, and $\boldsymbol{\epsilon}_m = [\epsilon_1 \quad \epsilon_2 \quad \epsilon_3]^{\mathrm{T}}$ is a vector representing the imaginary part of the quaternion. $\boldsymbol{h}_h,\ \boldsymbol{h}_i\ \in \mathbb{R}^6$ are wrench vectors exerted on the master manipulator and slave manipulators, respectively. $\boldsymbol{h}_h = [\boldsymbol{f}_h^{\mathrm{T}}\ \ \boldsymbol{t}_h^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{h}_i = [\boldsymbol{f}_i^{\mathrm{T}}\ \ \boldsymbol{t}_i^{\mathrm{T}}]^{\mathrm{T}}$, where $\boldsymbol{f}_*$ and $\boldsymbol{t}_*$ denote the force and torque, respectively. To correspond to the dimension of the wrench vectors, we define

$$\dot{\boldsymbol{x}}_m = [\dot{\boldsymbol{p}}_m^{\mathrm{T}}\ \ \boldsymbol{\omega}_m^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6, \quad \ddot{\boldsymbol{x}}_m = [\ddot{\boldsymbol{p}}_m^{\mathrm{T}}\ \ \dot{\boldsymbol{\omega}}_m^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6,$$

where $\boldsymbol{\omega}_m \in \mathbb{R}^3$ and $\dot{\boldsymbol{\omega}}_m \in \mathbb{R}^3$ are angular velocity and acceleration, which can be resolved from (3.20). Accordingly, we have $\bar{M}_m,\ \bar{M}_i,\ \in \mathbb{R}^{6\times 6}$, $\bar{C}_m,\ \bar{C}_i\ \in \mathbb{R}^{6\times 6}$, $\bar{\boldsymbol{g}}_m,\ \bar{\boldsymbol{g}}_i\ \in \mathbb{R}^6$, $\boldsymbol{u}_m,\ \boldsymbol{u}_i\ \in \mathbb{R}^6$, and $\boldsymbol{\rho}_m,\ \boldsymbol{\rho}_i\ \in \mathbb{R}^6$.

For the motion regulation in the slave side, the reference trajectory sent from the master side is geometrically distributed into multiple desired trajectories in task space for the according slave manipulators. As well, in a cooperative task, the wrench exerted by the object is required to be well distributed for the manipulators ensemble. To allow potentially heterogeneous payload capacities and internal wrenches compensation [62, 70, 73], an optimal force allocation strategy will be used in this chapter.

As discussed in Chapter 6, it is optional to feedback the slave motion signals to the master side depending on the controller requirements. Force signals, however, are usually required in teleoperation systems to enhance the telepresence of the human operator. In this study, we choose to send back the environmental force exerted on the manipulator along with the load distribution matrix so that the operator can perceive either the environmental wrench that exerts on the leader only or that exerted on the overall slave system. The force estimation and prediction, as in Chapter 6 will be used to predict the environmental wrench for the master manipulator as well as the human operator on the master side.

Without loss of generality, the following assumptions are made for the subsequent discussions.

Assumption 7.1. *The manipulated object is a rigid and undeformable body. Also, it is tightly grasped by the end effectors of the slave manipulators such that we can determine the motion and load allocation through the geometrical distribution of the manipulators.*

Assumption 7.2. *The pose and velocity of the end effectors are measurable, while the* *acceleration signals are obtained by time differentiation. Besides, the environmental force and human force input are assumed measurable.*

Assumption 7.3. *Network delays are time-varying as modelled in (3.31), and the average delays between the master and the slave are considerably longer than that those amongst the slaves.*

Assumption 7.4. *The weight of the object is beyond one manipulator's capacity but below the sum of the involving manipulators' maximum payload capacity. In addition, through the load allocation strategy, the distributed load is within the payload capacity of the manipulators ensemble.*

Control Objective

The objective is to drive pose ($\boldsymbol{x}_{\mathrm{O}}$) of an object handled by multiple slave manipulators towards a reference pose trajectory commanded by a master manipulator, that is $\|\boldsymbol{x}_{\mathrm{O}}(t) - \boldsymbol{x}_i(t)\|_1 \leq \epsilon_{ms}$ with a small bound $\epsilon_{ms} \geq 0$. Controllers to be designed should be capable of dealing with the unknown modelling information (including parametric uncertainties, unknown frictions, and external disturbances) and unknown time-varying network delays. A motion and load distribution strategy will be designed to allocate the desired motion and wrench for the slave manipulators ensemble. Additionally, the allocation approach should be able to compensate for the additional torque caused by the heterogeneous wrench distribution. On the master side, the environmental wrench is to be predicted with a small prediction error.

7.2 Controller Development

In what follows, we present the controller design for the master, leader, and slave manipulators. The wrench allocation strategy will be introduced as well.

7.2.1 Master Controller Design

The master manipulator is directly handled by the human operator on the local side. To allow the motion controller design, a target impedance model is designed to generate reference motions in task space. An environmental wrench predictor based on recursive estimation is used to recover the wrench from the transmitted environmental wrench parameters, and, meanwhile, reduce the effect of network delays.

Target Impedance Model

Procedures to design the master controller are similar to Section 6.2.2. Nevertheless, due to the use of the unit quaternion to represent the orientation vector, the pose vector has a different dimension to other motion (e.g., velocity vectors) and wrench vectors. Therefore, in view of Hogan [110], the target impedance model is redesigned in joint space as

$$\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d + M_r(\boldsymbol{q}_d - \boldsymbol{q}_r) + C_r J^{\mathrm{T}}(\boldsymbol{q}_r)(\boldsymbol{h}_h - \hat{\boldsymbol{h}}_0), \tag{7.3}$$

where $\hat{\boldsymbol{h}}_0 \in \mathbb{R}^6$ is the estimation of the environmental wrench exerted on the leader manipulator in the slave side. $J(\boldsymbol{q}_r) \in \mathbb{R}^{6\times 3}$ is the Jacobian matrix given in (3.23). $[\boldsymbol{q}_d^{\mathrm{T}} \;\; \dot{\boldsymbol{q}}_d^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ is the desired trajectory in joint space. From (7.3), we obtain the reference signal in joint space, i.e., $[\boldsymbol{q}_r^{\mathrm{T}} \;\; \dot{\boldsymbol{q}}_r^{\mathrm{T}} \;\; \ddot{\boldsymbol{q}}_r^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^9$, which is, then, used to calculate the reference signal for the master end effector as $\boldsymbol{x}_r = [\boldsymbol{p}_r^{\mathrm{T}} \;\; \boldsymbol{\xi}_r^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^7$, $\dot{\boldsymbol{x}}_r = [\dot{\boldsymbol{p}}_r^{\mathrm{T}} \;\; \boldsymbol{\omega}_r^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$, and $\ddot{\boldsymbol{x}}_r = [\ddot{\boldsymbol{p}}_r^{\mathrm{T}} \;\; \dot{\boldsymbol{\omega}}_r^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ through the forward kinematics as described in Section 3.2.

Remark 7.1. *[111] The impedance parameters $M_r \in \mathbb{R}^{3\times 3}$ and $C_r \in \mathbb{R}^{3\times 3}$ must be carefully selected such that the impedance model is not subject to unstable behaviour and the target impedance is attainable for the manipulators.*

Environmental Wrench Prediction

In order to avoid the direct transmission of power signals over the network channels, the environmental wrench is transformed into non-power signals that are actually transmitted. In the slave side, let $\Phi_{0,e} \in \mathbb{R}^{6\times 3}$ denote the known regressor as

$$\Phi_{0,e} = \begin{bmatrix} \dot{\boldsymbol{p}}_0(1) & \dot{\boldsymbol{p}}_0(2) & \dot{\boldsymbol{p}}_0(3) & \boldsymbol{\omega}_0(1) & \boldsymbol{\omega}_0(2) & \boldsymbol{\omega}_0(3) \\ \boldsymbol{p}_0(1) & \boldsymbol{p}_0(2) & \boldsymbol{p}_0(3) & \boldsymbol{x}_{\xi,0}(1) & \boldsymbol{x}_{\xi,0}(2) & \boldsymbol{x}_{\xi,0}(3) \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^{\mathrm{T}}. \tag{7.4}$$

Note that since the orientation is represented by unit quaternion that is a 4-tuple, in (7.4) we use $\boldsymbol{x}_{\xi,0} = \eta_0 \boldsymbol{\epsilon}_0 \in \mathbb{R}^3$, where $\boldsymbol{\xi}_0 = [\eta_0 \;\; \boldsymbol{\epsilon}_0^{\mathrm{T}}]^{\mathrm{T}}$ is the orientation of the leader end effector. To facilitate the following expression of the wrench estimation, it is denoted that[3] $\Phi_{0,e}\{k\} = [\dot{\boldsymbol{x}}_0(k) \quad \boldsymbol{X}_0(k) \quad 1]^{\mathrm{T}} \in \mathbb{R}^3$, where $\boldsymbol{X}_0 = [\boldsymbol{p}_0^{\mathrm{T}} \quad \boldsymbol{x}_{\xi,0}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{6\times 1}$ and $k = 1, 2, \cdots, 6$. Consider the linearization in (6.4), the environmental wrench can be represented by an environment wrench parameter $\Theta_{0,e}\{k\} \in \mathbb{R}^{3\times 1}$ [107] which is recursively estimated by

$$\begin{aligned} \dot{\hat{\Theta}}_{0,e}\{k\} &= -H_0\Phi_{0,e}\{k\}(\Phi_{0,e}^T\{k\}\hat{\Theta}_{0,e}\{k\} - \boldsymbol{h}_0(k)), && (7.5) \\ \frac{d}{dt}H_0^{-1} &= \Phi_{0,e}\{k\}\Phi_{0,e}^T\{k\}, && (7.6) \end{aligned}$$

where H_0 is initialized as $H_0 = h_0 I_{3\times 3}, \ h_0 > 0$. Then, the estimated environment wrench parameter $\hat{\Theta}_{0,e}\{k\}$ is sent to the master side to predict the environmental wrench $\hat{\boldsymbol{h}}_0$ by the following wrench predictor:

$$\hat{\boldsymbol{h}}_0\{k\} = \hat{\Theta}_{0,e}\{k\}(t - T_{0m}(t))\Phi_m\{k\}. \tag{7.7}$$

where $\hat{\Theta}_{0,e}\{k\}(t - T_{0m}(t))$ denotes the delayed signal of $\hat{\Theta}_{0,e}\{k\}$. $T_{0m}(t)$ represents the network delay of channels from the slave side to the master side. $\Phi_m\{k\} = [\dot{\boldsymbol{x}}_m(k), \boldsymbol{X}_{\xi,m}(k), 1]^T \in \mathbb{R}^{3\times 1}$, where $\boldsymbol{X}_m = [\boldsymbol{p}_m^{\mathrm{T}} \quad \boldsymbol{x}_{\xi,m}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^6$ and $\boldsymbol{x}_{\xi,m} = \eta_m \boldsymbol{\epsilon}_m \in \mathbb{R}^3$. $k = 1, 2, \cdots, 6$.

Master Controller

To design the ANTSM controller for the master manipulator, the pose tracking error is defined by

$$\left\{ \begin{aligned} \boldsymbol{e}_m^l &= \boldsymbol{p}_m - \boldsymbol{p}_r \\ \dot{\boldsymbol{e}}_m^l &= \dot{\boldsymbol{p}}_m - \dot{\boldsymbol{p}}_r \end{aligned} \right. \qquad \left\{ \begin{aligned} \boldsymbol{e}_m^\xi &= \boldsymbol{\xi}_m \otimes \boldsymbol{\xi}_r^* \\ \dot{\boldsymbol{e}}_m^a &= \boldsymbol{w}_m - \boldsymbol{w}_r, \end{aligned} \right. \tag{7.8}$$

where the superscript $*^l$ denotes the translational errors and $*^a$ the angular errors. $\otimes$ is the quaternion product defined in (3.17). $\boldsymbol{\xi}_r^*$ is the conjugate of reference orientation $\boldsymbol{\xi}_r$, i.e., $\boldsymbol{\xi}_r^* = [\eta_r \quad -\boldsymbol{\epsilon}_r^{\mathrm{T}}]^{\mathrm{T}}$. As the dimension of the quaternion error is $\boldsymbol{e}_m^\xi \in \mathbb{R}^4$, let $\boldsymbol{e}_m^a = \boldsymbol{e}_m^a(1)[\boldsymbol{e}_m^a(2) \quad \boldsymbol{e}_m^a(3) \quad \boldsymbol{e}_m^a(4)]^{\mathrm{T}} \in \mathbb{R}^3$. Consider the definition of quaternion error in (3.17), we have

$$\boldsymbol{e}_m^a = (\eta_m\eta_r + \boldsymbol{\epsilon}_m^T\boldsymbol{\epsilon}_r)(-\eta_m\boldsymbol{\epsilon}_r + \eta_r\boldsymbol{\epsilon}_m - S(\boldsymbol{\epsilon}_m)\boldsymbol{\epsilon}_r). \tag{7.9}$$

---

[3] In this chapter, the augment of $A\{k\}$ denotes a vector that is the k^{th} row of a matrix A, while $A(k)$ denotes the k^{th} element of a vector A.

Then, we can stack the according translational errors and angular errors as

$$\boldsymbol{e}_m = \begin{bmatrix} \boldsymbol{e}_m^l \\ \boldsymbol{e}_m^a \end{bmatrix} \in \mathbb{R}^6, \quad \dot{\boldsymbol{e}}_m = \begin{bmatrix} \dot{\boldsymbol{e}}_m^l \\ \dot{\boldsymbol{e}}_m^a \end{bmatrix} \in \mathbb{R}^6. \tag{7.10}$$

With the reference signals from (7.3), the ANTSM controller, $\boldsymbol{u}_m \in \mathbb{R}^6$, for the master manipulator is described as:

$$\begin{aligned} \boldsymbol{u}_m &= \bar{C}_m \dot{\boldsymbol{x}}_m + \bar{\boldsymbol{g}}_m - \boldsymbol{h}_h + \bar{M}_m \Big\{ \frac{-\dot{\boldsymbol{e}}_m^{(2-\alpha_m)}}{\alpha_m \beta_m} + \ddot{\boldsymbol{x}}_r \\ &\quad - \bar{M}_m^{M+} \hat{\Theta}_m \Phi_m \mathrm{sgn}(\boldsymbol{s}_m) - \kappa_m \mathrm{sgn}(\boldsymbol{s}_m) \Big\}, \qquad (7.11) \\ \dot{\hat{\Theta}}_m &= -\alpha_m \beta_m \| \boldsymbol{s}_m^T \mathrm{diag}[\dot{\boldsymbol{e}}_m^{(\alpha_m - 1)}] \|_1 \bar{M}_m^{M+} \Phi_m, \qquad (7.12) \end{aligned}$$

where $\boldsymbol{s}_m \in \mathbb{R}^6$ is the sliding surface given by $\boldsymbol{s}_m = \boldsymbol{e}_m + \beta_m (\dot{\boldsymbol{e}}_m^v)^{\alpha_m}$. $\kappa_m > 0$, $\bar{M}_m^{M+} \geq \|\bar{M}_m^+\|_1$ and $\hat{\Theta}_m$ is the estimate of Θ_m. The stability proof follows the same procedure as that of Theorem 6.1, which is provided in Appendix C.1.

7.2.2 Slave Controller Design

Controllers to be designed on the slave side include the leader controller and follower controllers. The selected leader manipulator receives the desired trajectory of the object commanded by the human operator and feedbacks the force signals to the master side. Meanwhile, the leader serves as an intermediary that passes the distributed information to the followers.

In synchronization tasks, given a time-varying reference pose, the goal is to simply drive the robots to track the poses with respect to each robot's own base coordinate systems and, thus, there is not force interaction involved amongst the manipulators. However, a cooperative manipulation system requires the robots' pose and payload to be well distributed for each individual manipulator. When a cooperative system consists of heterogeneous manipulators, the distributed poses and forces may vary from one manipulator to another due to their different load capacities. In what follows, an optimal wrench allocation approach will be introduced.

Optimal Wrench Allocation

Fig 7.2 illustrates a cooperative manipulation system with one leader and three followers that hold a common object. Some pose representations are defined as follows.

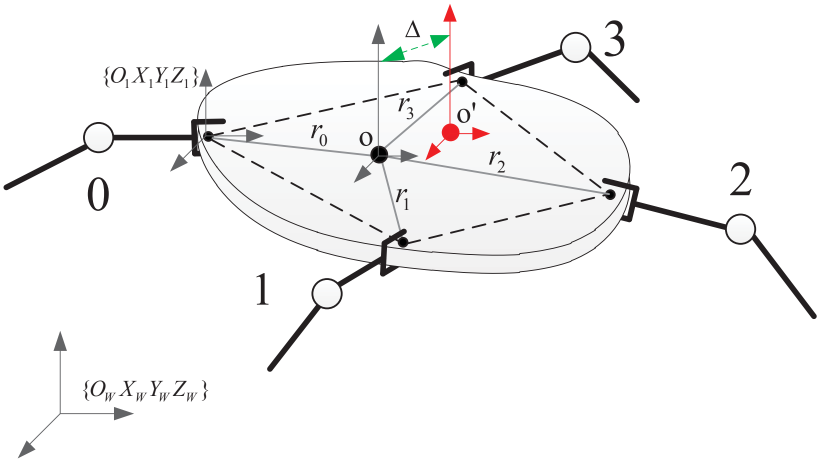


Figure 7.2: Grasp kinematics of the cooperative manipulation system.

- $\mathbf{O}$: center of mass (CoM) of the object.
- $\boldsymbol{p}_{\mathbf{O}} \in \mathbb{R}^3$: position of $\mathbf{O}$ with respect to the world frame $\{O_wX_wY_wZ_w\}$.
- $\boldsymbol{p}_i \in \mathbb{R}^3$: position of the grasp point of the i^{th} manipulator with respect to the world frame $\{O_wX_wY_wZ_w\}$. $(i = 0, 1, 2, \cdots, n)$.
- $\boldsymbol{\xi}_{\mathbf{O}} \in \mathbb{R}^4$: orientation (in unit-quaternion representation) of $\mathbf{O}$ with respect to the world frame $\{O_wX_wY_wZ_w\}$.
- $\boldsymbol{\xi}^i_{\mathbf{O}} \in \mathbb{R}^4$: orientation of frame $\{O_{\mathbf{O}}X_{\mathbf{O}}Y_{\mathbf{O}}Z_{\mathbf{O}}\}$ with respect to coordinate $\{O_iX_iY_iZ_i\}$ of the i^{th} grasp point.
- $\boldsymbol{\xi}_i \in \mathbb{R}^4$: orientation of the grasp point of the i^{th} manipulator with respect to the world frame $\{O_wX_wY_wZ_w\}$.
- $\boldsymbol{r}_i \in \mathbb{R}^3$: position of the CoM with respect to the coordinate $\{O_iX_iY_iZ_i\}$ of the i^{th} grasp point.
- $R^{\mathbf{O}}_i \in \mathbb{R}^{3\times3}$: rotation matrix of frame $\{O_iX_iY_iZ_i\}$ with respect to the frame $\{O_{\mathbf{O}}X_{\mathbf{O}}Y_{\mathbf{O}}Z_{\mathbf{O}}\}$.

The dynamics of the object in task space are given by

$$M_{\mathbf{O}}\ddot{\boldsymbol{x}}_{\mathbf{O}} + C_{\mathbf{O}}\dot{\boldsymbol{x}}_{\mathbf{O}} + \boldsymbol{g}_{\mathbf{O}} = \boldsymbol{h}_{\mathbf{O}}, \tag{7.13}$$

where the pose $\boldsymbol{x}_{\mathbf{O}} = [\boldsymbol{p}_{\mathbf{O}}^T \ \boldsymbol{\xi}_{\mathbf{O}}^T]^T \in \mathbb{R}^7$, $\dot{\boldsymbol{x}}_{\mathbf{O}} = [\dot{\boldsymbol{p}}_{\mathbf{O}}^T \ \boldsymbol{\omega}_{\mathbf{O}}^T]^T \in \mathbb{R}^6$, and $\ddot{\boldsymbol{x}}_{\mathbf{O}} = [\ddot{\boldsymbol{p}}_{\mathbf{O}}^T \ \dot{\boldsymbol{\omega}}_{\mathbf{O}}^T]^T \in \mathbb{R}^6$. $\boldsymbol{h}_{\mathbf{O}} \in \mathbb{R}^6$ is the task space wrench that exerts on the slave end effectors. Parameters of the object dynamics are given as

$$M_{\mathbf{O}} = \begin{bmatrix} m_{\mathbf{O}} I_3 & 0_3 \\ 0_3 & I_{\mathbf{O}} \end{bmatrix}, \ C_{\mathbf{O}} = \begin{bmatrix} 0_3 & 0_3 \\ 0_3 & \omega_{\mathbf{O}} I_{\mathbf{O}} \end{bmatrix}, \ \boldsymbol{g}_{\mathbf{O}} = \begin{bmatrix} -m_{\mathbf{O}} g \\ \mathbf{0}_{3\times 1} \end{bmatrix}, \tag{7.14}$$

where $m_{\mathbf{O}}$ and $I_{\mathbf{O}} \in \mathbb{R}^{3\times 3}$ are the object's mass and inertia, respectively (Note that I with a number subscript denotes an identity matrix of the appropriate dimension).

Remark 7.2. *The desired motion of the object is commanded by the master manipulator, so we have $\boldsymbol{x}_{\boldsymbol{O}} = \boldsymbol{x}_m(t - T_{m0}(t))$, $\dot{\boldsymbol{x}}_{\boldsymbol{O}} = \dot{\boldsymbol{x}}_m(t - T_{m0}(t))$, and $\ddot{\boldsymbol{x}}_{\boldsymbol{O}} = \ddot{\boldsymbol{x}}_m(t - T_{m0}(t))$. Then, from (7.13), the wrench exerted on the slave manipulators can be expressed as*

$$\boldsymbol{h}_{\boldsymbol{O}} = M_{\boldsymbol{O}} \ddot{\boldsymbol{x}}_m(t - T_{m0}(t)) + C_{\boldsymbol{O}} \dot{\boldsymbol{x}}_m(t - T_{m0}(t)) + \boldsymbol{g}_{\boldsymbol{O}}. \tag{7.15}$$

The slave manipulators' end effectors are subject to the following constraints:

$$\boldsymbol{p}_i^d = \boldsymbol{p}_{\mathbf{O}} + R_i^{\mathbf{O}} \boldsymbol{r}_i, \tag{7.16}$$

$$\boldsymbol{\xi}_i^d = \boldsymbol{\xi}_{\mathbf{O}}^i \otimes \boldsymbol{\xi}_{\mathbf{O}}. \tag{7.17}$$

For the velocity distribution, let $\dot{\boldsymbol{x}}_i^d = [(\dot{\boldsymbol{p}}_i^d)^{\mathrm{T}} \ (\boldsymbol{\omega}_i^d)^{\mathrm{T}}]^{\mathrm{T}}$ define the allocated desired velocities for the slave manipulators. By stacking the desired slave velocities into one vector, we have $\dot{\boldsymbol{x}}_d = [(\dot{\boldsymbol{x}}_0^d)^{\mathrm{T}} \ (\dot{\boldsymbol{x}}_1^d)^{\mathrm{T}} \ \cdots \ (\dot{\boldsymbol{x}}_n^d)^{\mathrm{T}}]^{\mathrm{T}}$. A grasp matrix G is used to describe the velocity constraint [70] as:

$$\dot{\boldsymbol{x}}_d = G^T \dot{\boldsymbol{x}}_{\mathbf{O}}, \tag{7.18}$$

$$G = \begin{bmatrix} I_3 & 0_3 & \cdots & I_3 & 0_3 \\ S(\boldsymbol{r}_0) & I_3 & \cdots & S(\boldsymbol{r}_n) & I_3 \end{bmatrix}. \tag{7.19}$$

In contrast, provided the measured pose, $\boldsymbol{x}_i = [\boldsymbol{p}_i^{\mathrm{T}}, \ \boldsymbol{\xi}_i^{\mathrm{T}}]^{\mathrm{T}}$ of the manipulator i, the pose of the object, $\boldsymbol{x}_{\mathbf{Oi}} = [\boldsymbol{p}_{\mathbf{Oi}}^{\mathrm{T}} \ \boldsymbol{\xi}_{\mathbf{Oi}}^{\mathrm{T}}]^{\mathrm{T}}$, can be obtained by

$$\boldsymbol{p}_{\mathbf{O}i} = \boldsymbol{p}_i - R_i^{\mathbf{O}} \boldsymbol{r}_i, \tag{7.20}$$

$$\boldsymbol{\xi}_{\mathbf{O}i} = (\boldsymbol{\xi}_{\mathbf{O}}^i)^{-1} \otimes \boldsymbol{\xi}_i. \tag{7.21}$$

Note that $\boldsymbol{x}_{\mathbf{Oi}}$ may not be equal to $\boldsymbol{x}_{\mathbf{O}}$ as in (7.13) because of the tracking error of the manipulators towards the distributed motions.

Remark 7.3. *Consider that a subset of follower manipulators are not directly connected to the leader, they are driven to follow their connected neighbours' trajectories that are transformed into those with respect to these followers' own base frames. For example, the neighbor state* $\boldsymbol{x}_j$ *is transformed to* $\boldsymbol{x}_j^i = [(\boldsymbol{p}_j^i)^T \quad (\boldsymbol{\xi}_j^i)^T]^T$ *that is with respect to the base frame of the* i^{th} *manipulator and given by*

$$\begin{aligned} \boldsymbol{p}_i^j &= \boldsymbol{p}_j - R_j^{\boldsymbol{O}} \boldsymbol{r}_j + R_i^{\boldsymbol{O}} \boldsymbol{r}_i, & (7.22) \\ \boldsymbol{\xi}_i^j &= \boldsymbol{\xi}_{\boldsymbol{O}}^i \otimes (\boldsymbol{\xi}_j^{\boldsymbol{O}})^{-1} \otimes \boldsymbol{\xi}_j. & (7.23) \end{aligned}$$

As to the load distribution, given the payload, $\boldsymbol{h}_{\mathbf{O}}$, of the manipulated object, and let a stacked vector $\boldsymbol{h} = [(\boldsymbol{h}_0)^{\mathrm{T}} \quad (\boldsymbol{h}_1)^{\mathrm{T}} \quad \cdots, (\boldsymbol{h}_n)^{\mathrm{T}}]^{\mathrm{T}}$ denote the distribute wrenches for the slave manipulators, the grasp matrix G in (7.19) relates the distributed wrenches in $\boldsymbol{h}^d$ and the object wrench $\boldsymbol{h}_{\mathbf{O}}$ as:

$$\boldsymbol{h}_{\mathbf{O}} = G\boldsymbol{h} = \begin{bmatrix} I_3 & 0_3 & \cdots & I_3 & 0_3 \\ S(\boldsymbol{r}_0) & I_3 & \cdots & S(\boldsymbol{r}_n) & I_3 \end{bmatrix} \boldsymbol{h}. \tag{7.24}$$

However, The equality in (7.24) holds only when the fixed point O exactly locates at the CoM (see Fig.7.2), which is not always true in reality. For example, when the cooperative system is constructed by multiple heterogeneous manipulators with different load capacities, the force is not equally distributed, which generates additional torques exerted on the object's physical center of mass. Then, it results in a non-zero displacement, Δ, between the virtual mass center (denoted by O′) and the physical center, as shown in Fig.7.2. With this in mind, the following Lemma [71] provides an optimized load distribution strategy for situations where the physical CoM does not coincide with the virtual CoM.

Lemma 7.1. *To compensate the additional torques that cause the non-zero displacement,* Δ*, between the physical CoM and the virtual CoM, a set of distribution coefficients* ζ_i *are introduced such that* $\sum_{i=1}^n \zeta_i = 1$*. Then, the optimized load distribution is described by a quadratic minimization of the wrench as*

$$min\, \{\boldsymbol{h}^T W \boldsymbol{h}\}, \quad s.t. \;\; \boldsymbol{h}_{\boldsymbol{O}} = G\boldsymbol{h}, \tag{7.25}$$

where $W = blockdiag\{\frac{1}{\zeta_i} I_6\}$ *and* I_6 *is a* 6×6 *identity matrix. Then, the optimization modifies the load distribution strategy as*

$$\boldsymbol{h} = G^{+}\boldsymbol{h}_o + C_\lambda \begin{bmatrix} S(\Delta) K_\Delta \\ S(\Delta) \end{bmatrix} \boldsymbol{h}_o, \tag{7.26}$$

where $\Delta = \sum_{i=1}^{n} \zeta_i \boldsymbol{r}_i$ *is the displacement as shown in Fig.7.2.* G^+ *denotes the generalized inverse of the grasp matrix* G *and* C_λ *is a compensation matrix given by*

$$G^+ = \begin{bmatrix} \zeta_1 I_3 & \zeta_1 S(r_1) K_\Delta \\ 0_3 & \zeta_1 K_\Delta \\ \vdots & \vdots \\ \zeta_n I_3 & \zeta_n S(r_n) K_\Delta \\ 0_3 & \zeta_n K_\Delta \end{bmatrix}, \quad C_\lambda = \begin{bmatrix} \zeta_1 I_3 & \lambda_1 S_{\lambda 1} K_\lambda \\ 0_3 & \lambda_1 K_\lambda \\ \vdots & \vdots \\ \zeta_n I_3 & \lambda_n S_{\lambda n} K_\lambda \\ 0_3 & \lambda_n K_\lambda \end{bmatrix}.$$

λ_i *is a torque compensation coefficient in the case where* $\Delta \neq \boldsymbol{0}$. $K_\Delta = [I_3 + \sum_{i=1}^{n} \zeta_i S^T(\boldsymbol{r}_i) S(\boldsymbol{r}_i) - S^T(\Delta) S(\Delta)]^{-1}$, $K_\lambda = [I_3 + \sum_{i=1}^{n} \lambda_i S^T(\boldsymbol{r}_i) S(\boldsymbol{r}_i) - S^T(\Delta_\lambda) S(\Delta_\lambda)]^{-1}$, $S_{\lambda i} = S(\Delta_\lambda) - S(r_i)$ *and* $\Delta_\lambda = \sum_{i=1}^{n} \lambda_i \boldsymbol{r}_i$.

Remark 7.4. *Eq.(7.26) provides a more general solution for the load distribution in a cooperative manipulation system. The load distribution strategy in (7.24) is a special case where the wrench is equally distributed to the manipulators ensemble, and the displacement is zero, i.e.,* $\Delta = 0$.

With the distributed motion and wrench information for the slave manipulators, their controllers can be designed in a distributed manner.

Leader Controller Design

The leader manipulator is driven to follow the allocated signal $\boldsymbol{x}_0^d = [(\boldsymbol{p}_0^d)^{\mathrm{T}} \ \ (\boldsymbol{\xi}_0^d)^{\mathrm{T}}]^{\mathrm{T}}$ and $\dot{\boldsymbol{x}}_0^d = [(\dot{\boldsymbol{p}}_0^d)^{\mathrm{T}} \ \ (\boldsymbol{\omega}_0^d)^{\mathrm{T}}]^{\mathrm{T}}$. Since the desired pose $\boldsymbol{x}_0^d$ is calculated from the network-delayed master signals $\boldsymbol{x}_m(t - T_{m0}(t))$, the mixed-type feedback signals, $\boldsymbol{e}_0 \in \mathbb{R}^6$ and $\dot{\boldsymbol{e}}_0 \in \mathbb{R}^6$, are used to compute the leader controller.

$$\begin{cases} \boldsymbol{e}_0 = \begin{bmatrix} \boldsymbol{e}_0^l \\ \boldsymbol{e}_0^a \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_0(t - \bar{T}_1) - \boldsymbol{p}_0^d, \\ (\eta_0 \eta_0^d + \boldsymbol{\epsilon}_0^T \boldsymbol{\epsilon}_0^d)(-\eta_0 \boldsymbol{\epsilon}_0^d + \eta_0^d \boldsymbol{\epsilon}_0 - S(\boldsymbol{\epsilon}_0) \boldsymbol{\epsilon}_0^d) \end{bmatrix} \\ \dot{\boldsymbol{e}}_0 = \begin{bmatrix} \dot{\boldsymbol{e}}_0^l \\ \dot{\boldsymbol{e}}_0^a \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_0(t) - \dot{\boldsymbol{p}}_0^d \\ \boldsymbol{w}_0(t) - \boldsymbol{\omega}_0^d) \end{bmatrix}, \end{cases} \tag{7.27}$$

where $\bar{T}_1$ is the average delay of $T_{m0}(t)$. Note that, in (7.27), $\boldsymbol{\xi}_0 = [\eta_0 \ \ \boldsymbol{\epsilon}_0^{\mathrm{T}}]^{\mathrm{T}}$ is actually a self-delayed signal as $\boldsymbol{\xi}_0(t - \bar{T}_1) = [\eta_0(t - \bar{T}_1) \ \ \boldsymbol{\epsilon}_0^{\mathrm{T}}(t - \bar{T}_1)]^{\mathrm{T}}$. The functional dependency of time and delay in (7.27) is omitted for compactness.

Based on the results in (4.14) and (7.27), an auxiliary self-delayed velocity error $\dot{\boldsymbol{\varepsilon}}_0^{sd} \in \mathbb{R}^6$ can be given by

$$\dot{\boldsymbol{\varepsilon}}_0^{sd} = \begin{bmatrix} \dot{\boldsymbol{e}}_0^{sd,l} \\ \dot{\boldsymbol{e}}_0^{sd,a} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_0(t) - \dot{\boldsymbol{p}}_0(t-\bar{T}_1) \\ \boldsymbol{w}_0(t) - \boldsymbol{w}_0(t-\bar{T}_1)] \end{bmatrix}. \tag{7.28}$$

Let $\mathbb{B}_0^{sd}$ denote the upper bound of $\dot{\boldsymbol{\varepsilon}}_0^{sd}$, i.e., $\|\dot{\boldsymbol{\varepsilon}}_0^{sd}\|_1 \leq \mathbb{B}_0^{sd}$.

The sliding surface of $\boldsymbol{s}_0 \in \mathbb{R}^6$ for the leader control design is given by

$$\boldsymbol{s}_0 = \boldsymbol{e}_0 + \beta_0 \dot{\boldsymbol{e}}_0^{\alpha_0}, \tag{7.29}$$

and the ANTSM controller. $\boldsymbol{u}_0 \in \mathbb{R}^6$, for the leader manipulator is designed as

$$\begin{aligned} \boldsymbol{u}_0 \;\; = \;\; & \boldsymbol{h}_0 + \bar{C}_0 \dot{\boldsymbol{x}}_0 + \bar{\boldsymbol{g}}_0 + \bar{M}_0 \Big\{ \frac{-\dot{\boldsymbol{e}}_0^{(2-\alpha_0)}}{\alpha_0 \beta_0} + \ddot{\boldsymbol{x}}_0^d \\ & -\frac{1}{\alpha_0\beta_0} \mathbb{B}_0^{sd} \mathrm{diag}(\dot{\boldsymbol{e}}_0^{(1-\alpha_0)}) \mathrm{sgn}(\boldsymbol{s}_0) - \bar{M}_0^{M+} \hat{\Theta}_0 \Phi_0 \mathrm{sgn}(\boldsymbol{s}_0) - \kappa_0 \mathrm{sgn}(\boldsymbol{s}_0) \Big\}, \end{aligned} \tag{7.30}$$

where $\hat{\Theta}_0$ is the estimate of Θ_0 updated by the adaptive law as

$$\dot{\hat{\Theta}}_0 = -\alpha_0 \beta_0 \|\boldsymbol{s}_0^T \mathrm{diag}[\dot{\boldsymbol{e}}_0^{(\alpha_0-1)}]\|_1 \bar{M}_0^{M+} \Phi_0. \tag{7.31}$$

The stability proof follows the same steps as in Appendix C.2.

Follower Controller Design

We consider that the slave manipulators are network connected and experience delays. Therefore, the mixed-type feedback is used in the subsequent control design.

$$\begin{cases} \boldsymbol{e}_i = \begin{bmatrix} \boldsymbol{e}_i^l \\ \boldsymbol{e}_i^a \end{bmatrix} & = \sum_{j=1}^n a_{ij} \begin{bmatrix} \boldsymbol{p}_i(t-\bar{T}_2) - \boldsymbol{p}_i^j(t-T_{ij}(t)), \\ (\eta_i \eta_i^j + \boldsymbol{\epsilon}_i^T \boldsymbol{\epsilon}_j)(-\eta_i \boldsymbol{\epsilon}_i^j + \eta_i^j \boldsymbol{\epsilon}_i - S(\boldsymbol{\epsilon}_i)\boldsymbol{\epsilon}_i^j) \end{bmatrix} \\ & \quad + b_i \begin{bmatrix} \boldsymbol{p}_i(t-\bar{T}_1-\bar{T}_2) - \boldsymbol{p}_i^d(t-T_{i0}(t)), \\ (\eta_i' \eta_i^d + \boldsymbol{\epsilon}_i'^T \boldsymbol{\epsilon}_i^d)(-\eta_i' \boldsymbol{\epsilon}_i^d + \eta_i^d \boldsymbol{\epsilon}_i' - S(\boldsymbol{\epsilon}_i')\boldsymbol{\epsilon}_i^d) \end{bmatrix}, \\ \dot{\boldsymbol{e}}_i = \begin{bmatrix} \dot{\boldsymbol{e}}_i^l \\ \dot{\boldsymbol{e}}_i^a \end{bmatrix} & = \sum_{j=1}^n a_{ij} \begin{bmatrix} \dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{p}}_i^j(t-T_{ij}(t)) \\ \boldsymbol{w}_i(t) - \boldsymbol{w}_i^j(t-T_{ij}(t)) \end{bmatrix} \\ & \quad + b_i \begin{bmatrix} \dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{p}}_i^d(t-T_{i0}(t)) \\ \boldsymbol{w}_i(t) - \boldsymbol{w}_i^d(t-T_{i0}(t)) \end{bmatrix}, \end{cases} \tag{7.32}$$

where $\boldsymbol{e}_i \in \mathbb{R}^6$ and $\dot{\boldsymbol{e}}_i \in \mathbb{R}^6$ are the position error and velocity error, respectively. $\boldsymbol{x}_i^j = [(\boldsymbol{p}_i^j)^{\mathrm{T}},\ (\boldsymbol{\xi}_i^j)^{\mathrm{T}}]^{\mathrm{T}}$ is the poses of neighbor manipulators that are transformed to those with respect to the base frame of the agent i as described in Remark 7.3. $T_{ij}(t)$ and $T_{i0}(t)$ are the time-varying network delays from agent j to i, and agent 0 to i, respectively. $\bar{T}_2$ is the average delay of $T_{ij}(t)$ and $T_{i0}(t)$. Again, in (7.32), $\boldsymbol{\xi}_i = [\eta_i\ \ \boldsymbol{\epsilon}_i^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{\xi}_i' = [\eta_i'\ \ \boldsymbol{\epsilon}_i'^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{\xi}_i^j = [\eta_i^j\ \ \boldsymbol{\epsilon}_i^{j\mathrm{T}}]^{\mathrm{T}}$, and $\boldsymbol{\xi}_i^d = [\eta_i^d\ \ \boldsymbol{\epsilon}_i^{d\mathrm{T}}]^{\mathrm{T}}$ are actually delayed signals that are accordingly expressed as

$$\begin{aligned}
\boldsymbol{\xi}_i(t-\bar{T}_2) &= [\eta_i(t-\bar{T}_2)\ \ \boldsymbol{\epsilon}_i^{\mathrm{T}}(t-\bar{T}_2)]^{\mathrm{T}}, \\
\boldsymbol{\xi}_i'(t-\bar{T}_1-\bar{T}_2) &= [\eta_i(t-\bar{T}_1-\bar{T}_2)\ \ \boldsymbol{\epsilon}_i^{\mathrm{T}}(t-\bar{T}_1-\bar{T}_2)]^{\mathrm{T}}, \\
\boldsymbol{\xi}_i^j(t-T_2(t)) &= [\eta_i^j(t-T_2(t))\ \ \boldsymbol{\epsilon}_i^{j\mathrm{T}}(t-T_2(t))]^{\mathrm{T}}, \\
\boldsymbol{\xi}_i^d(t-T_2(t)) &= [\eta_i^d(t-T_2(t))\ \ \boldsymbol{\epsilon}_i^{d\mathrm{T}}(t-T_2(t))]^{\mathrm{T}}.
\end{aligned}$$

These functional dependencies of time and delays are omitted for compactness in (7.32).

Similar to the results in (7.28), we can find an auxiliary self-delayed velocity error $\dot{\boldsymbol{\varepsilon}}_i^{sd} \in \mathbb{R}^6$ to establish the relationship between the position error and velocity as in (7.32), that is,

$$\begin{aligned}
\dot{\boldsymbol{\varepsilon}}_i^{sd} &= \begin{bmatrix} \dot{\boldsymbol{e}}_i^{sd,l} \\ \dot{\boldsymbol{e}}_i^{sd,a} \end{bmatrix} \\
&= \sum_{j=1}^{n} a_{ij} \begin{bmatrix} \dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{p}}_i(t-\bar{T}_2) \\ \boldsymbol{w}_i(t) - \boldsymbol{w}_i(t-\bar{T}_2)] \end{bmatrix} + b_i \begin{bmatrix} \dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{p}}_i(t-\bar{T}_1-\bar{T}_2) \\ \boldsymbol{w}_i(t) - \boldsymbol{w}_i(t-\bar{T}_1-\bar{T}_2)] \end{bmatrix}. \quad (7.33)
\end{aligned}$$

Let $\mathbb{B}_i^{sd}$ denote the upper bound of $\dot{\boldsymbol{\varepsilon}}_i^{sd}$ such that $\|\dot{\boldsymbol{\varepsilon}}_i^{sd}\|_1 \leq \mathbb{B}_i^{sd}$. The sliding surface of $\boldsymbol{s}_i \in \mathbb{R}^6$ is given by

$$\boldsymbol{s}_i = \boldsymbol{e}_i + \beta_i \dot{\boldsymbol{e}}_i^{\alpha_i}, \quad (7.34)$$

and follower controller, $\boldsymbol{u}_i \in \mathbb{R}^6$, is designed as

$$\begin{aligned}
\boldsymbol{u}_i &= \bar{C}_i \dot{\boldsymbol{x}}_i + \bar{\boldsymbol{g}}_i + \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{-1} \bar{M}_i \Bigg\{ \frac{-\dot{\boldsymbol{e}}_i^{(2-\alpha_i)}}{\alpha_i \beta_i} + \sum_{j=1}^{n} a_{ij} \ddot{\boldsymbol{x}}_i^j (t - T_{ij}) \\
&\quad + b_i \ddot{\boldsymbol{x}}_i^d (t - T_{i0}) - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big) \bar{M}_i^{M+} \hat{\Theta}_i \Phi_i \mathrm{sgn}(\boldsymbol{s}_i) \\
&\quad - \frac{1}{\alpha_i \beta_i} \mathbb{B}_i^{sd} \mathrm{diag}(\dot{\boldsymbol{e}}_i^{(1-\alpha_i)}) \mathrm{sgn}(\boldsymbol{s}_i) - \kappa_i \mathrm{sgn}(\boldsymbol{s}_i) \Bigg\} + \boldsymbol{h}_i, \quad (7.35)
\end{aligned}$$

where $\boldsymbol{h}_i$ is the distributed wrench given in Lemma 7.1. $\kappa_i > 0$ and $\bar{M}_i^{M+} \geq \|\bar{M}_i^{+}\|_1$. $\dot{\boldsymbol{e}}_i^{(\alpha-1)}$ follows the same operation as in (4.18). $\hat{\Theta}_i$ is the estimate of Θ_i and obeys the following adaptive law in dealing with the unknown modeling parameters.

$$\dot{\hat{\Theta}}_i = -\alpha_i\beta_i\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\|\boldsymbol{s}_i^T \mathrm{diag}[\dot{\boldsymbol{e}}_i^{(\alpha_i-1)}]\|_1 \bar{M}_i^{M+}\Phi_i. \tag{7.36}$$

The stability proof of the controller (7.35) and (7.36) follows the similar procedures as in Appendix B.1 and Appendix B.2.

In motion control, the tracking errors tend to cause the relative displacement between the end effector and object. However, since the slave manipulators' end effectors are assumed to rigidly attach to the object, tracking control errors give rise to the internal wrench exerted on the object. To assess the tracking error induced internal wrench, we assume that the contact of the end effector and object surface behaves like a point-contact-with-friction (PCWF) model [112]. Then, the internal wrench is approximated by:

$$\boldsymbol{h}_{int} = \frac{1}{2}\begin{bmatrix} K_{stiff}^{f}\Big[\sum_{i=0}^{n}(\boldsymbol{p}_i^{\mathrm{W}} - \boldsymbol{p}_i^{d,\mathrm{W}})\Big]^2 \\ K_{stiff}^{t}\Big[(\eta_0^{\mathrm{W}}\boldsymbol{\epsilon}_0^{\mathrm{W}} - \eta_0^{d,\mathrm{W}}\boldsymbol{\epsilon}_0^{d,\mathrm{W}}) - \sum_{i=1}^{n}(\eta_i^{\mathrm{W}}\boldsymbol{\epsilon}_i^{\mathrm{W}} - \eta_i^{d,\mathrm{W}}\boldsymbol{\epsilon}_i^{d,\mathrm{W}})\Big]^2 \end{bmatrix}, \tag{7.37}$$

where $K_{stiff}^{f} > 0$ and $K_{stiff}^{t} > 0$ denote the connecting stiffness for force and torque, respectively, between the end effector and the object. Variables with the superscript of $*^{\mathrm{W}}$ denote the transformed variables with respect to the world frame.

7.3 Simulation Results

The numerical simulations were performed with one master and two slave manipulators (one leader and one follower) using the mathematical model of the 3-DOF Phantom Omni haptic devices described in Section 3.3.3. Fig. 7.3 illustrates the setup of the SMMS manipulation system.

Unless otherwise specified, the initial setting for the manipulators is as follows. The manipulator's joints are initially static, and their positions are initialized as (unit: *rad*): $\boldsymbol{q}_m(0) = [-0.3\ \ 0.3\ \ -0.8]^{\mathrm{T}}$, $\boldsymbol{q}_0(0) = [0.7\ \ 0.6\ \ -0.5]^{\mathrm{T}}$, and $\boldsymbol{q}_1(0) = [0.3\ \ 0.6\ \ -1.1]^{\mathrm{T}}$. The desired joint signal is given as $\boldsymbol{q}_d(0) = [0.08\sin(\pi/10)\ \ 0.02\sin(\pi/10 + 0.7)\ \ -0.09\sin(\pi/10 - 1)]^{\mathrm{T}}$ *rad* for the calculation of reference signals in (7.3).

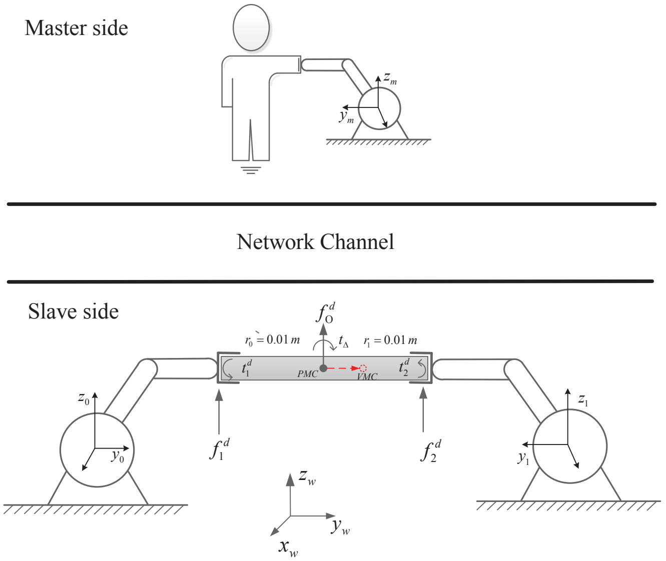


Figure 7.3: Schematic of the SMMS manipulation system using the Phantom Omni haptic devices.

A human wrench exerted on the master manipulator is given by: $[0.03\sin(ft) + 0.1\ \ 0.07\ \ 0.04\sin(ft)\ \ 0\ \ 0\ \ 0]^{\mathrm{T}}$ with $f = \pi/30$.

The random time-varying network delay is given as $T_{m0}(t) = T_{0m}(t) = 0.3 \pm 0.002\ s$ for teleoperation network channels and $T_{ij}(t) = T_{i0}(t) = 0.03 \pm 0.002\ s$ for cooperation network channels in the slave side. The same control gains were chosen for all agents' controllers as $\alpha_i = 7/5$ and $\beta_i = 1$, $\kappa_i = 13$. The frictional force coefficients are $\gamma_{m,s}^{1} = 3, \gamma_{m,s}^{2} = 6, \gamma_{m,s}^{3} = 1, \gamma_{m,s}^{4} = 2, \gamma_{m,s}^{5} = 4, \gamma_{m,s}^{6} = 1$. Fig.6.7 illustrates an example of the random network delays, external disturbance, and friction force.

Assume that the parameter uncertainties are 20% of the true values, that is, $\Delta M_{m,i} = 0.2M_{m,i}$, $\Delta C_{m,i} = 0.2C_{m,i}$, and $\Delta \boldsymbol{g}_{m,s} = 0.2\boldsymbol{g}_{m,s}$. $M_r = \mathrm{diag}[15\ 15\ 21]$ and $C_r = \mathrm{diag}[1\ 1\ 1]$ as in (7.3). $m_{\mathrm{O}} = 0.45$, $I_{\mathrm{O}} = \mathrm{diag}[0.5\ 0.5\ 0.5]$, and $g = 9.81$ as in (7.14). As discussed previously, in the sliding mode controllers, a saturation function replaces the sign function to reduce the chattering effect.

7.3.1 Performance of the Master Manipulator

The master manipulator's performance is evaluated by predicting the accuracy of the remote environmental wrench and the motion tracking accuracy of the master manipulator towards the reference signals. It is assumed that no torque generated

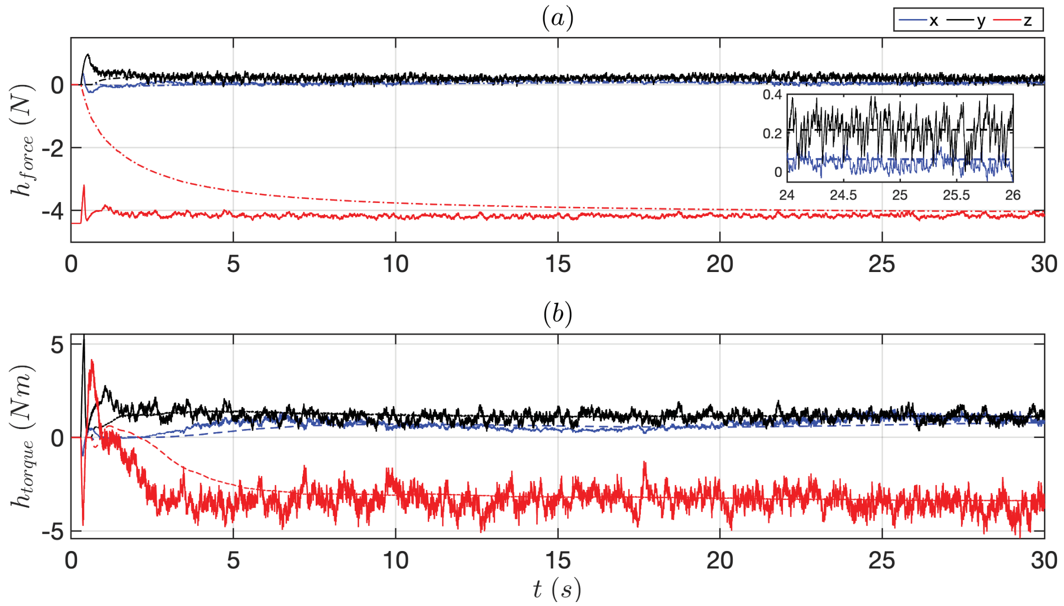


Figure 7.4: Environmental wrench prediction and comparison to the true values where the solid lines represent the actual environment forces measured in the slave side and dash lines represent the according predicted forces in the master side.

from the object model in (7.13), and the object force and its prediction are illustrated in Fig. 7.4. It shows that given the noisy environmental wrench, the force predictor provides the smooth estimations on the master side. The smoothness is due to the recursive process in the wrench estimation. This is beneficial in many cases as it provides the human operator with a better tactile sense when operating. However, using the recursive process cannot well predict the fast-changing and time-varying environmental wrench.

Fig.7.5 and Fig.7.6 respectively demonstrate the asymptotic translational and angular position synchronization of the master towards the generated reference signals. The designed master controller guarantees the finite-time tracking convergence (in about 3 s) and stability, despite the inexact information of system parameters, external noise bounds, and frictional forces. Tracking errors given by the difference between the reference trajectories and those of the master manipulator's end effector are presented in Fig.7.5 and Fig.7.6 as well. The tracking errors appear to be within small bounds during the steady-state phase. For example, the translational error bounds are in $[-3.179,\ -0.105] \times 10^{-3}\ m$, and $[-3.684,\ 4.167] \times 10^{-3}$ for the orientation tracking.

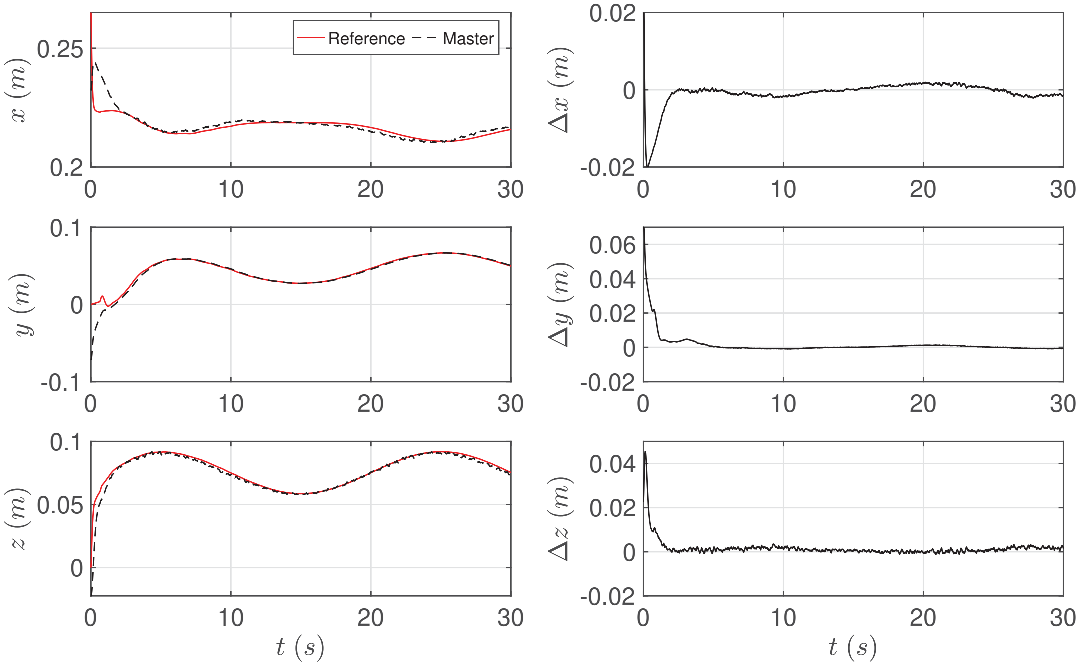


Figure 7.5: Translational position tracking of the master end effector towards the generated reference signals.

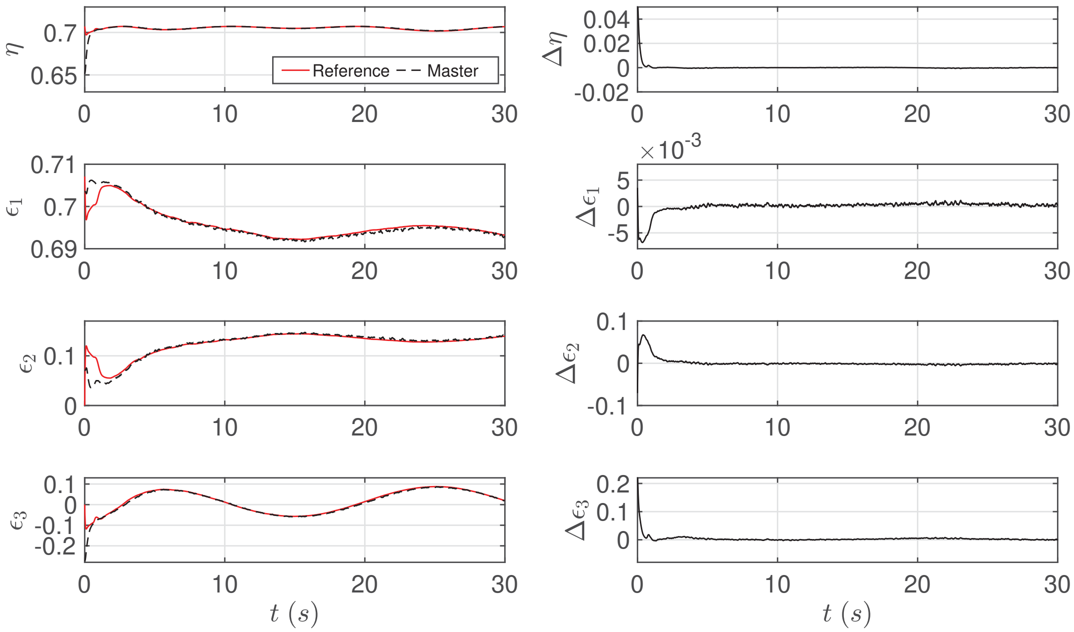


Figure 7.6: Angular position tracking of the master end effector towards the generated reference signals.

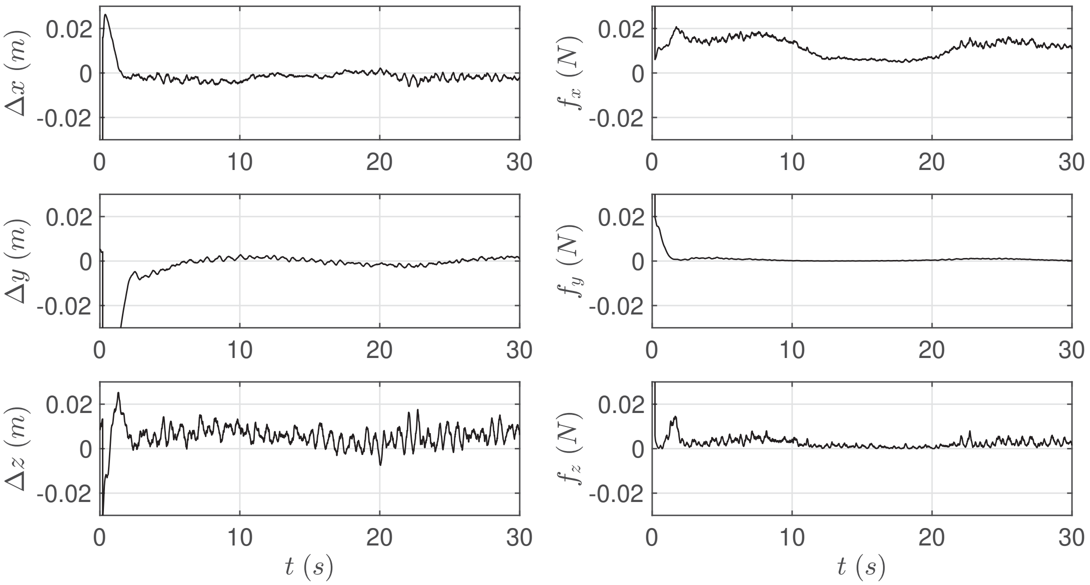


Figure 7.7: Position tracking errors and the resultant internal forces.

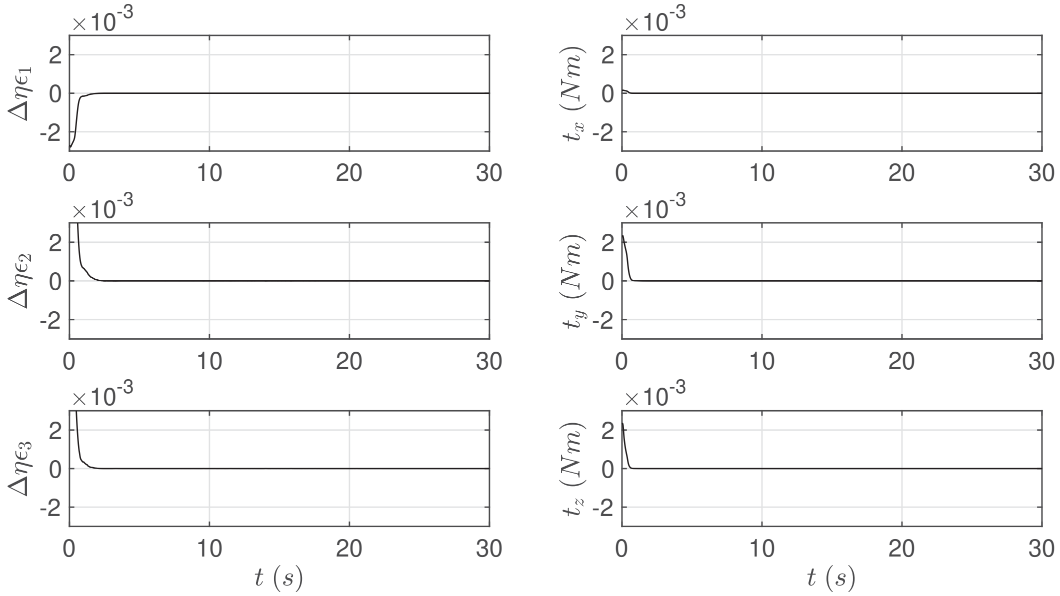


Figure 7.8: Orientation controller errors and the resultant internal torques.

7.3.2 Performance of the Cooperative Slave Manipulators

As formulated in (7.37), the tracking errors that tend to cause the relative displacement between the end effector and object actually appears in the form of internal force exerted on the object. By selecting the connecting stiffness as $K^f_{stiff} = 150$ and

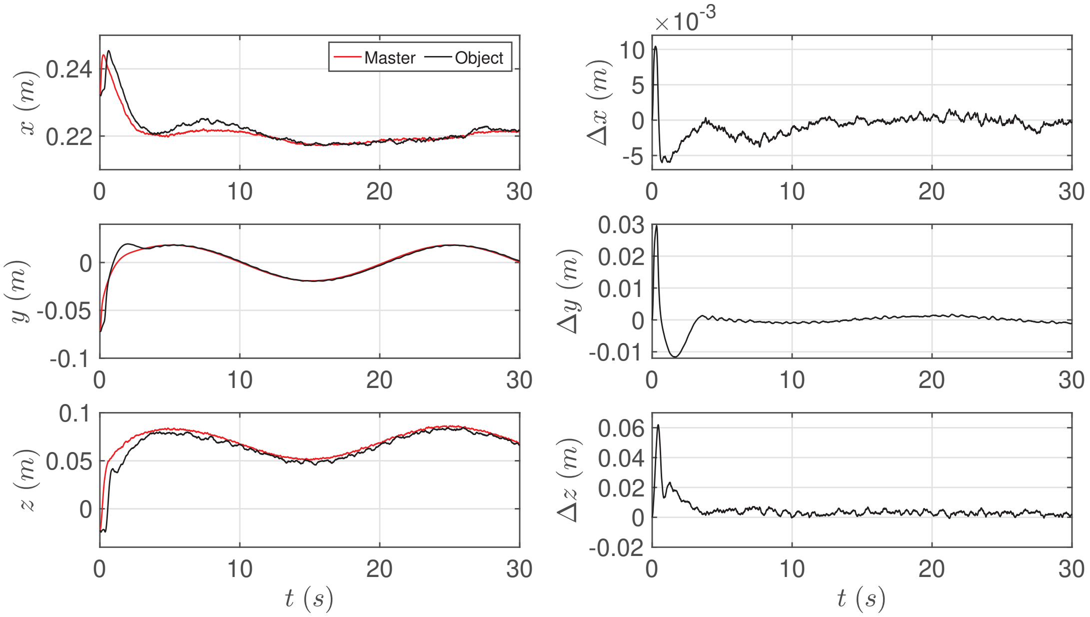


Figure 7.9: Translational position tracking of the leader end effector towards the master manipulator's translational trajectory.

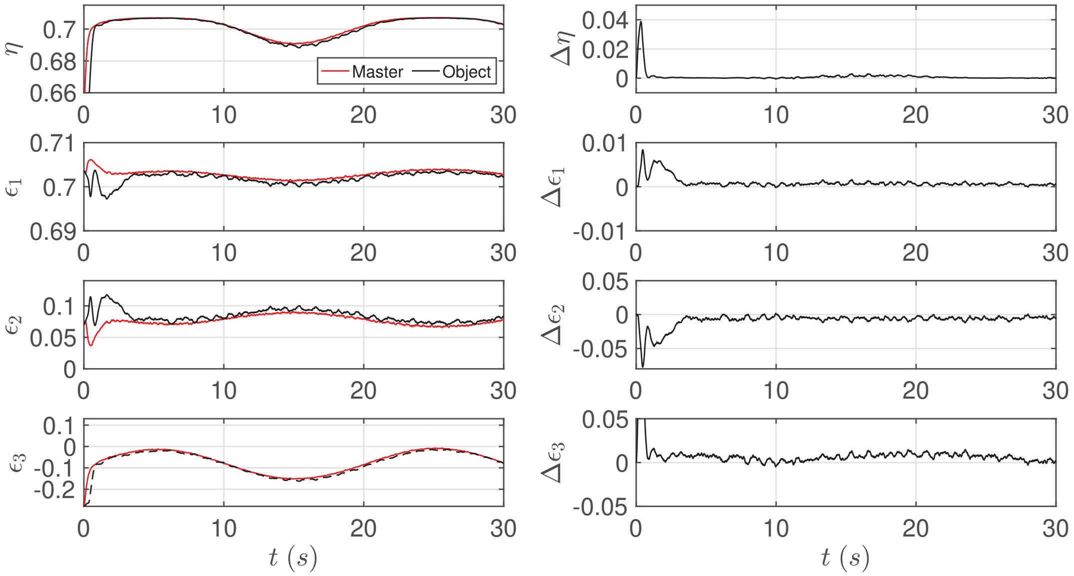


Figure 7.10: Angular position tracking of the leader end effector towards the master manipulator's orientation.

$K^t_{stiff} = 10$ [112], Fig.7.7 and Fig.7.8 illustrate the sum of pose tracking errors and their resultant internal wrench. The ANTSM controller provides accurate orientation control so that, in Fig.7.8, there is almost no internal torque about the x, y, and z

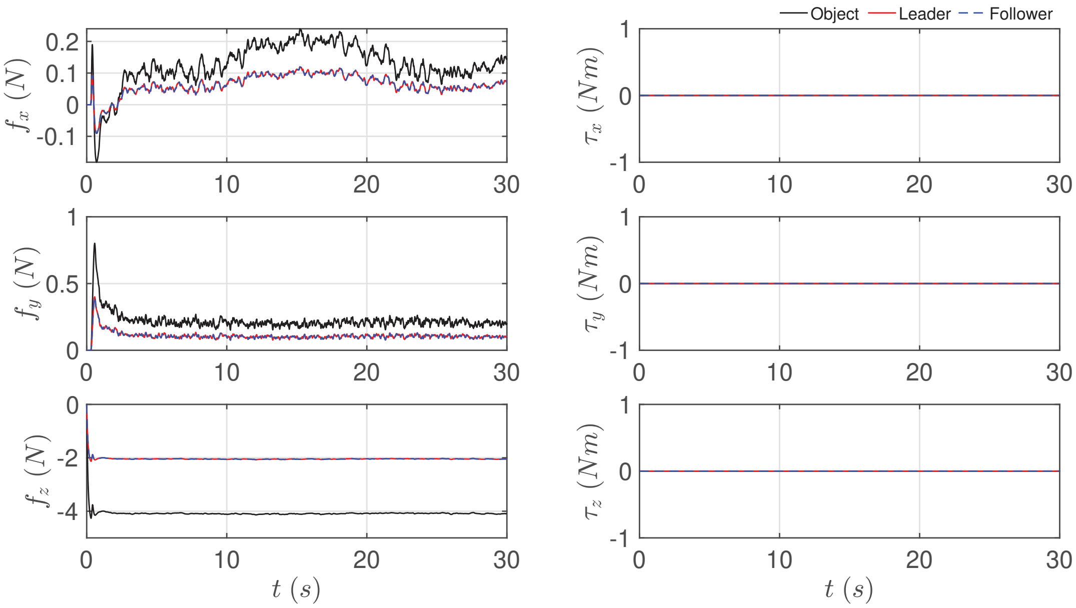


Figure 7.11: Wrench distribution with the weighting factors given in **Case 1**.

axes exerted on the object. Internal forces, however, are non-zero due to the translational tracking errors, as shown in Fig.7.7. Nevertheless, the internal forces along the the x, y, and z axes are very small, i.e., $\leq 0.02\ N$.

The tracking performance of the slave manipulators towards the master manipulator's states is presented by the pose tracking of the object towards the leader manipulator, which is shown in Fig.7.9 and Fig.7.10. As can be seen, the tracking synchronization can be achieved in a finite time (about 5 s). However, compared to the results in Fig.7.5 and Fig.7.6, the motion of the object presents more chatter. This is mainly caused by the effect of time delays. Nevertheless, the tracking error is still bounded and small. However, since we do not consider the time-varying control gain that is discussed in Chapter 5, we can see that the tracking errors during the transient phase are considerably large.

As to the load distribution, we assume that no torque is produced from the object dynamics as in (7.13). Two cases were simulated with the following weighting factors:

- **Case 1**: $\zeta_i = 0.5$ and $\lambda_i = 0.5$. $i = 1, 2$.
- **Case 2**: $\zeta_1 = 0.3$, $\zeta_2 = 0.7$, and $\zeta_i = \lambda_i$.

In **Case 1**, the selection of weighting factors results in the equal wrench distribution

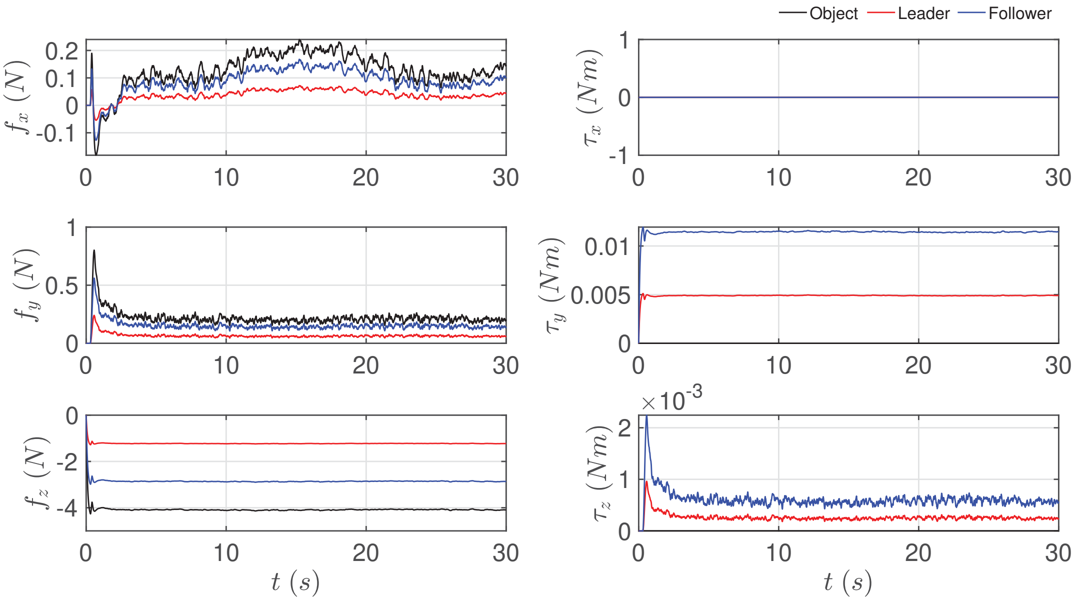


Figure 7.12: Wrench distribution with the weighting factors given in **Case 2**.

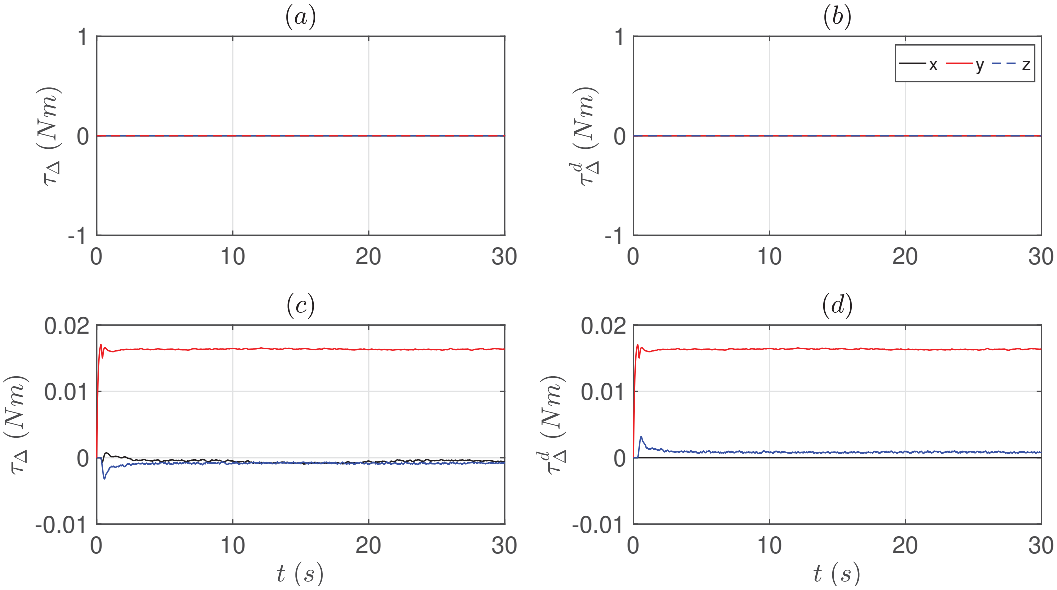


Figure 7.13: The produced additional torques in (a) **Case 1** and (b) **Case 2** are respectively compensated by the allocated torques shown in (c) and (d) from the wrench allocation policy.

to slave manipulators ensemble, and this is illustrated in Fig.7.11. The force exerted to the slave manipulators is primarily along z direction, because the force from the object is essentially caused by its gravity. As shown in Fig.7.11, the object force along the z-axis (around 4 N) is evenly distributed for the slave manipulators, i.e., 2 N for each agent. In addition, the selection of weighting factors in Case 1 implies that the coordinate frame O coincides with the center of mass of the object, which, in other words, the displacement $\Delta = 0$ as in Fig.7.2. As a result, there is no additional torque produced. Therefore, as illustrated in Fig.7.13 (a) and (b), the distributed desired torque and additional torques from the exerted forces are all zero.

In **Case 2**, the follower manipulator is assigned to bear more load than the leader manipulator. As show in Fig.7.12, additional torques are allocated about y-axis and z-axis. This is because the selection of weighting factors results in the offset coordinate frame O from the center of mass of the object, i.e., $\Delta \neq 0$. For example, the allocated forces along z-axis are $-1.232\ N$ for the leader manipulator and $-2.864\ N$ for the follower manipulator. As a result, the additional torque that the forces generate is $\tau_\Delta = r_1 \boldsymbol{f}_1(3) - r_2 \boldsymbol{f}_2(3) = 0.01 \times (-1.232) - 0.01 \times (-2.864)\ Nm = 0.016\ Nm$, as also shown in Fig.7.13 (d). Therefore, the torques in Fig.7.12 and Fig.7.13 (c) are allocated to compensate this additional torque such that no significant internal torque exerted on the object.

7.4 Concluding Remarks

In this chapter, the ANTSM control approaches have been successfully applied to a comprehensive SMMS manipulation system. Multiple manipulators in the slave side cooperatively handle a common object whose motion is commanded by a master manipulator. An optimal wrench distribution approach has been applied to allocate the motion and wrench to the slave manipulator ensemble. In addition, additional internal torques exerted on the object have been well compensated for by the allocated torques. Simulations were carried out with Phantom Omni haptic devices, and the results verified the guaranteed stability of the system and the good tracking accuracy that also helps reduce the internal wrench.

Chapter 8

Conclusions and Future Work

This chapter summarizes the contributions of this work and provides some future directions to extend the proposed control techniques for the networked cooperative manipulation systems and teleoperation systems.

8.1 Conclusions

In this thesis, concepts of the NTSM control method have been developed for a multi-manipulator system and its teleoperation. The goal of this work was to realize motion synchronization and force prediction with less effected by network constraints, modelling inaccuracy, and external disturbances.

By reviewing extensive existing literature in Chapter 2 and Chapter 1, we know that a semi-autonomous SMMS system enhances the human operators' ability to implement heavy and/or complex manipulation tasks in the remote and hazardous environments. However, the complexity of maintaining the tracking stability and accuracy arises when it comes to more real-life conditions, such as the network constraints, imperfect modelling of the robots, and nonlinear control requirements. These challenges have been well addressed by the developed ANTSM control method. The ANTSM controller has presented good adaptability and robustness in dealing with the parametric uncertainties, exogenous disturbances, and frictions without knowing their bounds. The stability of the proposed ANTSM control method has been proved by the Lyapunov functions. Numerical studies and experimental testing using the 3-DOF Phantom Omni haptic devices also showed the reliability and effectiveness of the developed ANTSM control strategy. The network constraints, including time-varying network delay, packet loss, and weakly connected topology, give rise to the phase-shift effect in the motion synchronization. This effect has been addressed by the proposed novel mixed-type feedback in the ANTSM controllers. The mixed-type feedback contains the non-self-delayed velocity that helps maintain the stability and the self-delayed position signal to reduce the tracking error. Sufficient conditions to attain the improvement of tracking accuracy are provided in Chapter 6 (i.e., Theorem 6.3). As the self-delayed process requires the knowledge of the network delay that is inexactly known and time-varying, the mean delay is used instead, while the random signals caused by uncertain delay deviation from its mean value can be regarded as signal disturbances that can be well dealt with by the robust control. Network delays considered in this work are up to 0.3 s, which is sufficient to represent the delays in many cooperation and teleoperation scenarios. The simulation results suggested significant improvements in reducing the phase shift effect.

A novel ANTSM time-varying switching gain as a piecewise logistic function was designed to address the issue where a constant control gain is unable to simultaneously achieve smooth convergence during the transient phase and the high tracking accuracy in the steady-state phase. The logistic function models a tunable exponential growth of the control gain with respect to time. Since the time of convergence to the desired trajectory can be known in the ANTSM controller method, one can adjust the evolution of the control gain to achieve a smooth convergence, and, meanwhile, improve the tracking accuracy during the steady-state phase, which was borne out by the experimental results. The tuning rules for time-varying switching gain have been generalized in Chapter 5 as well.

In multi-agent systems with heterogeneous robots, performance in terms of tracking accuracy, disturbance level, and payload capacity may vary from one robot to another. As a result, two particular topics, namely, neighbour selection and load distribution, should be taken into account. As to the neighbour selection, our theoretical studies have shown that unnecessarily using the information from the "less than perfect" neighbour/s can degrade an otherwise "well performing" agent's performance. An energy-based neighbour selection policy is introduced in Chapter 4 to help an agent efficiently identify and select proper neighbours in a synchronization task. Simulation results suggested that the proposed neighbour selection strategy successfully reduces the unnecessary interplay and the resulting energy accumulation of the overall system. As to the load distribution, an optimal wrench distribution approach has been applied to allocate the desired motion and wrench to the slave manipulators ensemble, as has been discussed in Chapter 7. The unwanted additional internal torques exerted on the object's center of mass have been well compensated by the allocated torques from the optimal wrench distribution approach. In addition, to transmit the wrench signals to the master side, an environmental wrench predictor provides the human operator with smooth and prompt wrench estimations regardless of the network delay effect.

Eventually, we built a comprehensive SMMS manipulation system where the human operator remotely commands the object's trajectory and perceives environmental wrench through the wrench predictor. On the slave side, multiple manipulators cooperatively handle a common object. Thanks to the target impedance model and distributed control framework, developments for the multi-agent systems and teleoperation systems have been successfully applied to the SMMS manipulation system. Simulations were carried out with three Phantom Omni haptic devices. The results verified not only the guaranteed stability of the system but also the good tracking accuracy that helps reduce the internal wrench exerted on the manipulated object.

8.2 Future Work

The developments given in this dissertation are fundamental, and there are many potential opportunities to continue the research. These are outside the scope of the work or were not explored due to constraints on space, resources or time.

From the theoretical point of view, some techniques for improving the ANTSM control method can be extended to other advanced control approaches when it comes to addressing similar issues. For example, the mixed-type feedback signals can be adopted in other linear or nonlinear control approaches to reduce the delay-induced phase shift effect. A detailed study of the effects of changing self delay on the resulting tracking performance would lead to much better insight in constructing the feedback signals. For control methods that require high control gain to obtain a better control accuracy yet experience significant overshoot/undershoot in the transient phase, using the control gain scheduled in a logistic growth manner would provide some exciting results of smooth convergence and high control accuracy. Also, since the wrench predictor used in this work only applies to constant wrench estimation, it would be beneficial to investigate other predictors that allow the prediction of the changing environmental wrench.

Another potential area for further research is the application of the robotic system. The proposed control theories have been verified and validated through numerical simulations and laboratory experiments under controlled conditions. Justifications of these developed theories could be further established if their industrial applicability could be further verified in the industrial setting. For example, remote control of a group of mobile manipulators with a transportation task. Manipulation performance is to be examined if more technical challenges (such as significant external disturbances from the terrain or changing weight of the tasks) could be posed to the control systems.

Our control strategies are developed with a general Euler Lagrange model that can be used to represent a wide variety of physical systems such as electrical, electromechanical and mechanical devices. Therefore, extending the developments to other types of robotic systems with different tasks is potential. For instance, in formation control of a group of semi-autonomous areal vehicles or ground vehicles, the ANTSM control schemes could be used to realize motion synchronization under network constraints and plant modelling uncertainties. Another example is the integration of robotic manipulators with other types of robots, such as mobile manipulators built from a robotic manipulator arm mounted on a mobile platform. Such systems combine the advantages of different robotic systems and reduce their drawbacks.

Bibliography

[1] M. W. Spong and M. Vidyasagar, *Robot dynamics and control.* John Wiley & Sons, 2008.

[2] Quanser, *2 DOF Serial Flexible Link.* `https://www.quanser.com/products/2-dof-serial-flexible-link/`.

[3] Quanser, *QArm 4 DOF serial robotic manipulator.* `https://www.quanser.com/products/qarm/`.

[4] U. Robots, *The UR3e collaborative robot.* `https://www.universal-robots.com/products/ur3-robot/`.

[5] Z. Erickson, H. M. Clever, G. Turk, C. K. Liu, and C. C. Kemp, "Deep haptic model predictive control for robot-assisted dressing," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 1–8, 2018.

[6] I. P. Releases, *Industrial Robots: Robot Investment Reaches Record 16.5 billion USD.* `https://ifr.org/news/robot-investment-reaches-record-16.5-billion-usd/`.

[7] S. Kock, T. Vittor, B. Matthias, H. Jerregard, M. Källman, I. Lundberg, R. Mellander, and M. Hedelind, "Robot concept for scalable, flexible assembly automation: A technology study on a harmless dual-armed robot," in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, pp. 1–5, 2011.

[8] A. Ryberg, M. Ericsson, A.-K. Christiansson, K. Eriksson, J. Nilsson, and M. Larsson, "Stereo vision for path correction in off-line programmed robot welding," in *2010 IEEE International Conference on Industrial Technology*, pp. 1700–1705, 2010.

[9] C. Connolly, "Technology and applications of abb robotstudio," *Industrial Robot: An International Journal*, 2009.

[10] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," Proceedings of American Control Conference, (Oregon, United States), pp. 1859–1864, June, 2005.

[11] A. S. Clair, A. Atrash, R. Mead, and M. J. Mataric, "Speech, gesture, and space: Investigating explicit and implicit communication in multi-human multi-robot collaborations.," in *AAAI Spring Symposium: Multirobot Systems and Physical Data Structures*, 2011.

[12] M. W. Carey, E. M. Kurz, J. D. Matte, T. D. Perrault, and T. Padir, "Novel EOD robot design with dexterous gripper and intuitive teleoperation," in *World Automation Congress 2012*, pp. 1–6, 2012.

[13] R. Saltaren, R. Aracil, C. Alvarez, E. Yime, and J. M. Sabater, "Field and service applications - Exploring deep sea by teleoperated robot - An underwater parallel robot with high navigation capabilities," *IEEE robotics & automation magazine*, vol. 14, no. 3, pp. 65–75, 2007.

[14] C. Preusche, T. Ortmaier, and G. Hirzinger, "Teleoperation concepts in minimal invasive surgery," *Control engineering practice*, vol. 10, no. 11, pp. 1245–1250, 2002.

[15] B. Siciliano and O. Khatib, *Springer handbook of robotics.* Springer, 2016.

[16] E. Coleshill, L. Oshinowo, R. Rembala, B. Bina, D. Rey, and S. Sindelar, "Dextre: Improving maintenance operations on the international space station," *Acta Astronautica*, vol. 64, no. 9-10, pp. 869–874, 2009.

[17] *Canadarm2, the Canadian robotic arm on the Space Station.* `https://www.asc-csa.gc.ca/eng/iss/canadarm2/default.asp`.

[18] R. Boumans and C. Heemskerk, "The European robotic arm for the international space station," *Robotics and Autonomous systems*, vol. 23, no. 1-2, pp. 17–27, 1998.

[19] S. Gidaro, M. Buscarini, E. Ruiz, M. Stark, and A. Labruzzo, "Telelap Alf-X: a novel telesurgical system for the 21st century.," *Surgical technology international*, vol. 22, pp. 20–25, 2012.

[20] J. Marescaux and F. Rubino, "The zeus robotic system: experimental and clinical applications," *Surgical Clinics*, vol. 83, no. 6, pp. 1305–1315, 2003.

[21] S. Maeso, M. Reza, J. A. Mayol, J. A. Blasco, M. Guerra, E. Andradas, and M. N. Plana, "Efficacy of the da vinci surgical system in abdominal surgery compared with that of laparoscopy: a systematic review and meta-analysis," 2010.

[22] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[23] E. W. Weisstein, "Weakly connected digraph," 1996. `http://mathworld.wolfram.com/WeaklyConnectedDigraph.html`, From MathWorld–A Wolfram Web Resource.

[24] E. Nuño, C. I. Aldana, and L. Basañez, "Task space consensus in networks of heterogeneous and uncertain robotic systems with variable time-delays," *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 6, pp. 917–937, 2017.

[25] R. Campa, K. Camarillo, and L. Arias, "Kinematic modeling and control of robot manipulators via unit quaternions: Application to a spherical wrist," Proceedings of 45th IEEE Conference on Decision and Control, (San Diego, California USA), pp. 6474–6479, December, 2006.

[26] C. K. Verginis, M. Mastellaro, and D. V. Dimarogonas, "Robust quaternion-based cooperative manipulation without force/torque information," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1754–1759, 2017.

[27] K. D. Hoang, F. Fioretto, W. Yeoh, E. Pontelli, and R. Zivan, "A large neighboring search schema for multi-agent optimization," in *International Conference on Principles and Practice of Constraint Programming*, pp. 688–706, Springer, 2018.

[28] Y. Mao and Z. Zhang, "Second-order consensus for multi-agent systems by state-dependent topology switching," in *2018 Annual American Control Conference (ACC)*, pp. 3392–3397, 2018.

[29] M. Zhihong, M. O'day, and X. Yu, "A robust adaptive terminal sliding mode control for rigid robotic manipulators," *Journal of Intelligent and Robotic systems*, vol. 24, no. 1, pp. 23–41, 1999.

[30] Y. Han and X. Liu, "Continuous higher-order sliding mode control with time-varying gain for a class of uncertain nonlinear systems," *ISA transactions*, vol. 62, pp. 193–201, 2016.

[31] W. Zhao, X. Ren, and S. Wang, "Parameter estimation-based time-varying sliding mode control for multimotor driving servo systems," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 5, pp. 2330–2341, 2017.

[32] D. Lee and M. W. Spong, "Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp. 360–365, 2005.

[33] O. M. Palafox and M. W. Spong, "Bilateral teleoperation of a formation of nonholonomic mobile robots under constant time delay," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2821–2826, 2009.

[34] M. Mohammadi, A. Franchi, D. Barcelli, and D. Prattichizzo, "Cooperative aerial tele-manipulation with haptic feedback," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5092–5098, 2016.

[35] K.-K. D. Young, "Controller design for a manipulator using theory of variable structure systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 2, pp. 101–109, 1978.

[36] J.-J. Slotine and S. S. Sastry, "Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators," *International journal of control*, vol. 38, no. 2, pp. 465–492, 1983.

[37] F. Harashima, H. Hashimoto, and K. Maruyama, "Practical robust control of robot arm using variable structure system," in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 532–539, 1986.

[38] H. Lee and V. I. Utkin, "Chattering suppression methods in sliding mode control systems," *Annual reviews in control*, vol. 31, no. 2, pp. 179–188, 2007.

[39] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding mode control and observation*. Springer, 2014.

[40] A. Ferrara and G. P. Incremona, "Design of an integral suboptimal second-order sliding mode controller for the robust motion control of robot manipulators," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2316–2325, 2015.

[41] S. Venkataraman and S. Gulati, "Control of nonlinear systems using terminal sliding modes," 1993.

[42] M. Zhihong, A. P. Paplinski, and H. R. Wu, "A robust mimo terminal sliding mode control scheme for rigid robotic manipulators," *IEEE transactions on automatic control*, vol. 39, no. 12, pp. 2464–2469, 1994.

[43] Y. Feng, X. Yu, and Z. Man, "Non-singular terminal sliding mode control of rigid manipulators," *Automatica*, vol. 38, no. 12, pp. 2159–2167, 2002.

[44] J.-J. Slotine and J. Coetsee, "Adaptive sliding controller synthesis for non-linear systems," *International Journal of Control*, vol. 43, no. 6, pp. 1631–1651, 1986.

[45] M. Park, O. Kwon, J. H. Park, S. Lee, and E. Cha, "Leader-following consensus criteria for multi-agent systems with time-varying delays and switching interconnection topologies," *Chinese Physics B*, vol. 21, no. 11, p. 110508, 2012.

[46] E. Nuño and R. Ortega, "Achieving consensus of euler–lagrange agents with interconnecting delays and without velocity measurements via passivity-based control," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 222–232, 2018.

[47] G. Ortega, F. Muñoz, E. E. Quesada, L. R. Garcia, and P. Ordaz, "Implementation of leader-follower linear consensus algorithm for coordination of multiple aircrafts," 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), (Krystal Cancun, Cancun, Mexico), pp. 25–32, November, 2015.

[48] Z. Meng, W. Ren, Y. Cao, and Z. You, “Leaderless and leader-following consensus with communication and input delays under a directed network topology,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 75–88, 2011.

[49] S. Liu, L. Xie, and F. L. Lewis, “Synchronization of multi-agent systems with delayed control input information from neighbors,” *Automatica*, vol. 47, no. 10, pp. 2152–2164, 2011.

[50] J. R. Klotz, S. Obuz, Z. Kan, and W. E. Dixon, “Synchronization of uncertain Euler–Lagrange systems with uncertain time-varying communication delays,” *IEEE transactions on cybernetics*, vol. 48, no. 2, pp. 807–817, 2018.

[51] E. Nuño, R. Ortega, L. Basanez, and D. Hill, “Synchronization of networks of nonidentical euler-lagrange systems with uncertain parameters and communication delays,” *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 935–941, 2011.

[52] U. Munz, A. Papachristodoulou, and F. Allgower, “Delay robustness in nonidentical multi-agent systems,” *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1597–1603, 2012.

[53] Y.-P. Tian and Y. Zhang, “High-order consensus of heterogeneous multi-agent systems with unknown communication delays,” *Automatica*, vol. 48, no. 6, pp. 1205–1212, 2012.

[54] H. Shen and Y.-J. Pan, “Tracking synchronization improvement of networked manipulators using novel adaptive non-singular terminal sliding mode control,” *IEEE Transactions on Industrial Electronics*, 2020.

[55] C.-L. Liu and Y.-P. Tian, “Formation control of multi-agent systems with heterogeneous communication delays,” *International Journal of Systems Science*, vol. 40, no. 6, pp. 627–636, 2009.

[56] R. Campa and H. De La Torre, “Pose control of robot manipulators using different orientation representations: A comparative review,” Proceedings of American Control Conference, (St. Louis, Missouri, USA), pp. 2855–2860, June, 2009.

[57] J. Luh, M. Walker, and R. Paul, “Resolved-acceleration control of mechanical manipulators,” *IEEE Transactions on Automatic Control*, vol. 25, no. 3, pp. 468–474, 1980.

[58] F. Caccavale, C. Natale, B. Siciliano, and L. Villani, “Six-DOF impedance control based on angle/axis representations,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 289–300, 1999.

[59] L. Ojeda and J. Borenstein, "Flexnav: Fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1, pp. 317–322, 2002.

[60] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE transactions on robotics and automation*, vol. 11, no. 3, pp. 328–342, 1995.

[61] H. Seraji, "Motion control of mobile manipulators," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, vol. 3, pp. 2056–2063, 1993.

[62] S. Erhart and S. Hirche, "Model and analysis of the interaction dynamics in cooperative manipulation tasks," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 672–683, 2016.

[63] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on automatic control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[64] H. Zhang, X. Zhou, Z. Wang, H. Yan, and J. Sun, "Adaptive consensus-based distributed target tracking with dynamic cluster in sensor networks," *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1580–1591, 2018.

[65] G. Xie and L. Wang, "Consensus control for networks of dynamic agents via active switching topology," in *International Conference on Natural Computation*, pp. 424–433, Springer, 2005.

[66] W. Yang, X. Wang, and H. Shi, "Optimal control nodes selection for consensus in multi-agent systems," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11697–11702, 2014.

[67] Y. Jia, N. Xi, and J. Buether, "Design of single-operator-multi-robot teleoperation systems with random communication delay," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 171–176, 2011.

[68] Y. Cheung and J. Chung, "Teleoperation of cooperative control of multiple heterogeneous slave unmanned aerial vehicles via a single master subsystem for multi-task multi-target scenarios," *SAE International Journal of Aerospace*, vol. 6, no. 2013-01-2105, pp. 139–149, 2013.

[69] Y. Cheung, J. H. Chung, and N. P. Coleman, "Semi-autonomous formation control of a single-master multi-slave teleoperation system," in *2009 IEEE Symposium on Computational Intelligence in Control and Automation*, pp. 117–124, 2009.

[70] T. E. Alberts and D. I. Soloway, “Force control of a multi-arm robot system,” in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pp. 1490–1496, 1988.

[71] A. Z. Bais, S. Erhart, L. Zaccarian, and S. Hirche, “Dynamic load distribution in cooperative manipulation tasks,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 2380–2385, 2015.

[72] I. D. Walker, R. A. Freeman, and S. I. Marcus, “Analysis of motion and internal loading of objects grasped by multiple cooperating manipulators,” *The International journal of robotics research*, vol. 10, no. 4, pp. 396–409, 1991.

[73] S. Erhart and S. Hirche, “Internal force analysis and load distribution for cooperative multi-robot manipulation,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1238–1243, 2015.

[74] U. Ahmad and Y.-J. Pan, “A time domain passivity approach for asymmetric multilateral teleoperation system,” *IEEE Access*, vol. 6, pp. 519–531, 2017.

[75] K. Haninger and M. Tomizuka, “Robust passivity and passivity relaxation for impedance control of flexible-joint robots with inner-loop torque control,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2671–2680, 2018.

[76] A. Franchi, P. R. Giordano, C. Secchi, H. I. Son, and H. H. Bülthoff, “A passivity-based decentralized approach for the bilateral teleoperation of a group of uavs with switching topology,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 898–905, 2011.

[77] E. Nuno, R. Ortega, N. Barabanov, and L. Basañez, “A globally stable pd controller for bilateral teleoperators,” *IEEE Transactions on Robotics*, vol. 24, no. 3, pp. 753–758, 2008.

[78] H. Shen, Y.-J. Pan, and B. He, “Teleoperation of multiple cooperative slave manipulators using graph-based non-singular terminal sliding-mode control,” in *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, pp. 1430–1435, 2017.

[79] H. Shen and Y.-J. Pan, “Improving tracking performance of nonlinear uncertain bilateral teleoperation systems with time-varying delays and disturbances,” *IEEE/ASME Transactions on Mechatronics*, 2019.

[80] Y.-C. Liu and M.-H. Khong, “Adaptive control for nonlinear teleoperators with uncertain kinematics and dynamics,” *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2550–2562, 2015.

[81] H. Jin, Z. Liu, H. Zhang, Y. Liu, and J. Zhao, “A dynamic parameter identification method for flexible joints based on adaptive control,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2896–2908, 2018.

[82] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control.* Springer Science & Business Media, 2010.

[83] S. Forbrigger, "Prediction-based haptic interfaces to improve transparency for complex virtual environments," 2017.

[84] T. R. Kurfess, *Robotics and automation handbook.* CRC press, 2018.

[85] C. Makkar, G. Hu, W. G. Sawyer, and W. E. Dixon, "Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction," *IEEE Transactions on Automatic Control*, vol. 52, no. 10, pp. 1988–1994, 2007.

[86] Z. Chen, Y.-J. Pan, and J. Gu, "Integrated adaptive robust control for multilateral teleoperation systems under arbitrary time delays," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 12, pp. 2708–2728, 2016.

[87] J. Hu and Y. Hong, "Leader-following coordination of multi-agent systems with coupling time delays," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 2, pp. 853–863, 2007.

[88] E. Nuño, L. Basañez, R. Ortega, and M. W. Spong, "Position tracking for non-linear teleoperators with variable time delay," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 895–910, 2009.

[89] K. Walker, Y.-J. Pan, and J. Gu, "Stochastic switching approach of bilateral teleoperation systems: Part II-experiment," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3737–3742, 2008.

[90] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell system technical journal*, vol. 39, no. 5, pp. 1253–1265, 1960.

[91] M. Ellis, D. P. Pezaros, T. Kypraios, and C. Perkins, "A two-level markov model for packet loss in UDP/IP-based real-time video applications targeting residential users," *Computer Networks*, vol. 70, pp. 384–399, 2014.

[92] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM Journal on Control and Optimization*, vol. 38, no. 3, pp. 751–766, 2000.

[93] H. Shen, Y.-J. Pan, U. Ahmad, and B. He, "Pose synchronization of multiple networked manipulators using nonsingular terminal sliding mode control," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.

[94] S. R. Munasinghe, M. Nakamura, S. Goto, and N. Kyura, "Trajectory planning for industrial robot manipulators considering assigned velocity and allowance under joint acceleration limit," *International Journal of Control, Automation, and Systems*, vol. 1, no. 1, pp. 68–75, 2003.

[95] R. Luck and A. Ray, “Delay compensation in integrated communication and control systems: Part I–conceptual development and analysis,” in *1990 American Control Conference*, pp. 2045–2050, 1990.

[96] R. Luck and A. Ray, “Delay compensation in integrated communication and control systems: Part ii–implementation and verification,” in *1990 American Control Conference*, pp. 2051–2055, 1990.

[97] S. Sarkka, “On unscented kalman filtering for state estimation of continuous-time nonlinear systems,” *IEEE Transactions on automatic control*, vol. 52, no. 9, pp. 1631–1641, 2007.

[98] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, “A new approach for filtering nonlinear systems,” in *Proceedings of 1995 American Control Conference-ACC'95*, vol. 3, pp. 1628–1632, 1995.

[99] T. Kontoroupi and A. W. Smyth, “Online noise identification for joint state and parameter estimation of nonlinear systems,” *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, vol. 2, no. 3, p. B4015006, 2015.

[100] J. C. Willems, “Dissipative dynamical systems part i: General theory,” *Archive for rational mechanics and analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[101] R. Ortega, J. A. L. Perez, P. J. Nicklasson, and H. J. Sira-Ramirez, *Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications*. Springer Science & Business Media, 2013.

[102] J. G. García, J. G. Ortega, A. S. García, and S. S. Martínez, “High-accuracy automatic system to assemble vehicle headlamps,” in *2009 IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–4, 2009.

[103] I. M. Shohet and Y. Rosenfeld, “Robotic mapping of building interior—precision analysis,” *Automation in construction*, vol. 7, no. 1, pp. 1–12, 1997.

[104] E. Gambao, C. Balaguer, and F. Gebhart, “Robot assembly system for computer-integrated construction,” *Automation in Construction*, vol. 9, no. 5-6, pp. 479–487, 2000.

[105] H. Shen and Y.-J. Pan, “Adaptive robust control of networked multi-manipulators with time-varying delays,” in *2019 American Control Conference (ACC)*, pp. 3670–3675, 2019.

[106] J. Liu and X. Wang, *Advanced sliding mode control for mechanical systems*. Springer, 2012.

[107] K. Yoshida and T. Namerikawa, “Stability and tracking properties in predictive control with adaptation for bilateral teleoperation,” in *2009 American Control Conference*, pp. 1323–1328, 2009.

[108] S. Khoo, L. Xie, and Z. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE/ASME transactions on mechatronics*, vol. 14, no. 2, pp. 219–228, 2009.

[109] H. Shen, Y.-J. Pan, and G. Bauer, "Manipulability-based load allocation and kinematic decoupling in cooperative manipulations," in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pp. 1168–1173, 2019.

[110] N. Hogan, "Impedance control: An approach to manipulation: Part ii—implementation," 1985.

[111] J. Koivumäki and J. Mattila, "Stability-guaranteed impedance control of hydraulic robotic manipulators," *IEEE/ASME TransactiOns On MechatrOnics*, vol. 22, no. 2, pp. 601–612, 2016.

[112] T. Wimboeck, C. Ott, and G. Hirzinger, "Passivity-based object-level impedance control for a multifingered hand," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4621–4627, 2006.

[113] S. Khoo, L. Xie, and Z. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE/ASME transactions on mechatronics*, vol. 14, no. 2, pp. 219–228, 2009.

[114] M. Zak, "Terminal attractors in neural networks," *Neural networks*, vol. 2, no. 4, pp. 259–274, 1989.

[115] "IEEE standard for transitions, pulses, and related waveforms," *IEEE Std 181-2011 (Revision of IEEE Std 181-2003)*, pp. 1–71, Sep. 2011.

[116] N. G. Paulter, D. R. Larson, and J. J. Blair, "The IEEE standard on transitions, pulses, and related waveforms, std-181-2003," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 4, pp. 1209–1217, 2004.

Appendix A

Theoratical Proof in Chapter 4

A.1 Proof of Theorem 4.1

Proof. After substituting the control input signal in (4.18) into (4.2), and cancelling out $C_i\dot{\boldsymbol{x}}_i + \boldsymbol{g}_i$, the closed-loop dynamics can be expressed as

$$\begin{aligned} M_i\ddot{\boldsymbol{x}}_i &= \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{-1} M_i\Bigg\{ -\frac{(\boldsymbol{e}_i^v)^{(2-\alpha)}}{\alpha\beta} + \frac{1}{\alpha\beta}\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(1-\alpha)}\Big)\dot{\boldsymbol{e}}_i^c \\ &\quad + \sum_{j=1}^{n} a_{ij}\ddot{\boldsymbol{x}}_j(t-T) + b_i\ddot{\boldsymbol{x}}_0(t-T) - \kappa_1\mathrm{sgn}(\boldsymbol{s}_i)\Bigg\}. \end{aligned}$$

Then, we have

$$\begin{aligned} &\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\ddot{\boldsymbol{x}}_i - \sum_{j=1}^{n} a_{ij}\ddot{\boldsymbol{x}}_j(t-T) - b_i\ddot{\boldsymbol{x}}_0(t-T) \\ &= -\frac{(\boldsymbol{e}_i^v)^{(2-\alpha)}}{\alpha\beta} + \frac{1}{\alpha\beta}\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(1-\alpha)}\Big)\dot{\boldsymbol{e}}_i^c - \kappa_1\mathrm{sgn}(\boldsymbol{s}_i). \qquad \text{(A.1)} \end{aligned}$$

As the right-hand side of (A.1) is equal to $\dot{\boldsymbol{e}}_i^v$, premultiplying by $\alpha\beta\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)$ on both sides leads to

$$\boldsymbol{e}_i^v + \alpha\beta\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)\dot{\boldsymbol{e}}_i^v - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\dot{\boldsymbol{e}}_i^c = -\alpha_i\beta_i\kappa_i\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)\mathrm{sgn}(\boldsymbol{s}_i).\text{(A.2)}$$

By using (4.17), (4.16), and (A.2), the time derivative of $\boldsymbol{s}_i$ can be derived as

$$\begin{aligned} \dot{\boldsymbol{s}}_i &= \dot{\boldsymbol{e}}_i^p + \alpha_i\beta_i\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)\dot{\boldsymbol{e}}_i^v \\ &= \boldsymbol{e}_i^v + \alpha_i\beta_i\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)\dot{\boldsymbol{e}}_i^v - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\dot{\boldsymbol{e}}_i^c \\ &= -\alpha_i\beta_i\kappa_i\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)\mathrm{sgn}(\boldsymbol{s}_i). \qquad \text{(A.3)} \end{aligned}$$

Consider the Lyapunov function candidate defined in a quadratic form as $V = \frac{1}{2}\boldsymbol{s}_i^T\boldsymbol{s}_i$, by using (A.3), the time derivative of V is determined as

$$\begin{aligned}\dot{V} &= \boldsymbol{s}_i^T\dot{\boldsymbol{s}}_i = \boldsymbol{s}_i^T[-\alpha_i\beta_i\kappa_i\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)\mathrm{sgn}(\boldsymbol{s}_i)] \\ &\leq -\sqrt{2}\alpha_i\beta_i\kappa_i(\|\boldsymbol{e}_i^v\|)^{(\alpha_i-1)}(\frac{1}{2}\boldsymbol{s}_i^T\boldsymbol{s}_i)^{\frac{1}{2}} \\ &= -\sqrt{2}\alpha_i\beta_i\kappa_i(\|\boldsymbol{e}_i^v\|)^{(\alpha_i-1)}V^{\frac{1}{2}}. \end{aligned} \tag{A.4}$$

Consider Lemma 3.3, we have $c = \sqrt{2}\alpha_i\beta_i\kappa_i(\|\boldsymbol{e}_i^v\|)^{(\alpha_i-1)}$ and $\gamma = \frac{1}{2}$. Therefore, when $\boldsymbol{e}_i^v \neq \boldsymbol{0}$, an estimate of the time T_i^s to reach the sliding surface $\boldsymbol{s}_i = \boldsymbol{0}$ is given by

$$T_i^s \leq \frac{(\boldsymbol{s}_i^T(0)\boldsymbol{s}_i(0))^{\frac{1}{2}}}{\alpha_i\beta_i\kappa_i\|\boldsymbol{e}_i^v\|^{\alpha_i-1}}, \tag{A.5}$$

where $\boldsymbol{s}_i(0) = \boldsymbol{e}_i^p(0) + \beta_i(\boldsymbol{e}_i^v(0))^{\alpha_i}$.

However, when $\boldsymbol{e}_i^v = \boldsymbol{0}$, (A.5) holds only when it can be proved that $\boldsymbol{e}_i^v = \boldsymbol{0}$ is not an attractor of $\boldsymbol{s}_i \neq \boldsymbol{0}$. By taking the time-derivative of the velocity error, the following expression can be obtained

$$\begin{aligned}\dot{\boldsymbol{e}}_i^v &= \sum_{j=1}^{n} a_{ij}[\ddot{\boldsymbol{x}}_i - \ddot{\boldsymbol{x}}_j(t-T)] + b_i[\ddot{\boldsymbol{x}}_i - \ddot{\boldsymbol{x}}_0(t-T)] \\ &= \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)[M_i^{-1}(\boldsymbol{u}_i - C\dot{\boldsymbol{x}}_i - \boldsymbol{g}_i)] - \sum_{j=1}^{n} a_{ij}\ddot{\boldsymbol{x}}_j(t-T) - b_i\ddot{\boldsymbol{x}}_0(t-T).\end{aligned} \tag{A.6}$$

Substituting the control input $\boldsymbol{u}_i$ in (4.18) into (A.6) leads to

$$\dot{\boldsymbol{e}}_i^v = \frac{1}{\alpha_i\beta_i}\mathrm{diag}\Big((\boldsymbol{e}_i^v)^{(\alpha-1)}\Big)\Big[-\boldsymbol{e}_i^v + \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\dot{\boldsymbol{e}}_i^c\Big] - \kappa_i\mathrm{sgn}(\boldsymbol{s}_i). \tag{A.7}$$

When $\boldsymbol{e}_i^v = \boldsymbol{0}$ and $\boldsymbol{s}_i \neq \boldsymbol{0}$, (A.7) becomes

$$\dot{\boldsymbol{e}}_i^v = -\kappa_i\mathrm{sgn}(\boldsymbol{s}_i). \tag{A.8}$$

Consider the fact that, for a vector $\boldsymbol{x} \in \mathbb{R}^{m\times 1}$, $(m = 1, 2, \cdots)$, $\|\boldsymbol{x}\| = 0$ if and only if $\boldsymbol{x} = \boldsymbol{0}$. Therefore, by the definition of $\mathrm{sgn}(\boldsymbol{s}_i)$, we know that $\|\mathrm{sgn}(\boldsymbol{s}_i)\| = 1$ when $\boldsymbol{s}_i \neq \boldsymbol{0}$. Therefore, $\|\dot{\boldsymbol{e}}_i^v\| = \kappa_i\|\mathrm{sgn}(\boldsymbol{s}_i)\| = \kappa_i > 0$, and thus we have that $\dot{\boldsymbol{e}}_i^v \neq \boldsymbol{0}$, which implies that $\boldsymbol{e}_i^v = \boldsymbol{0}$ is not an attractor [113] [114]. This complete the finiteness proof in (A.5).

Furthermore, as the numerator of the rational exponent $(1-\alpha_i)$ is always an even number, $(\boldsymbol{e}_i^v)^{1-\alpha_i} \geq \boldsymbol{0}$ always holds. Therefore, in (A.7), when the sliding surface $\boldsymbol{s}_i = \boldsymbol{0}$ is reached, we have

$$\begin{cases} \dot{\boldsymbol{e}}_i^v \leq \boldsymbol{0}, \;\; -\boldsymbol{e}_i^v + \Big(\sum_{j=1}^n a_{ij} + b_i\Big)\dot{\boldsymbol{e}}_i^c \leq \boldsymbol{0}, \\ \dot{\boldsymbol{e}}_i^v \geq \boldsymbol{0}, \;\; -\boldsymbol{e}_i^v + \Big(\sum_{j=1}^n a_{ij} + b_i\Big)\dot{\boldsymbol{e}}_i^c \geq \boldsymbol{0}. \end{cases} \tag{A.9}$$

The monotonicity in (A.9) suggests that there is an auxiliary switching surface given by

$$\|\boldsymbol{e}_i^v\| = \Big(\sum_{j=1}^n a_{ij} + b_i\Big)\|\dot{\boldsymbol{e}}_i^c\| \;\text{ s.t. }\; \operatorname{sgn}(\boldsymbol{e}_i^v) = \operatorname{sgn}(\dot{\boldsymbol{e}}_i^c) \tag{A.10}$$

that can always be reached. This concludes the proof. □

A.2 Proof of Proposition 4.1

Proof. For the translational self-delayed velocity error in (4.9), using the Schwartz inequality results in

$$\begin{aligned} \|\dot{\boldsymbol{e}}_i^{lc}\| &= \|\dot{\boldsymbol{p}}_i(t) - \dot{\boldsymbol{p}}_i(t-T)\| = \Big\| \int_0^T \ddot{\boldsymbol{p}}_i(\theta) d\theta \Big\| \\ &\leq \Big\| T \int_0^T |\ddot{\boldsymbol{p}}_i(t-\theta)|^2 d\theta \Big\|^{\frac{1}{2}} \leq T^{\frac{1}{2}} \|\ddot{\boldsymbol{p}}_i\| \leq T^{\frac{1}{2}} A^{lM}. \end{aligned} \tag{A.11}$$

Similarly, consider the angular self-delayed velocity error

$$\dot{\boldsymbol{e}}_i^{ac} = [\dot{\boldsymbol{\xi}}_i - \dot{\boldsymbol{\xi}}_i^*(t-T)] \otimes \boldsymbol{\xi}_j^*(t-T) + [\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^*(t-T)] \otimes \dot{\boldsymbol{\xi}}_j^*(t-T), \tag{A.12}$$

we have the $L2-$norm of $\dot{\boldsymbol{e}}_i^{ac}$ as

$$\begin{aligned} \|\dot{\boldsymbol{e}}_i^{ac}\| &= \Big\|[\dot{\boldsymbol{\xi}}_i - \dot{\boldsymbol{\xi}}_i^*(t-T)] \otimes \boldsymbol{\xi}_j^*(t-T)\Big\| + \Big\|[\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^*(t-T)] \otimes \dot{\boldsymbol{\xi}}_j^*(t-T)\Big\| \\ &= \|\dot{\boldsymbol{\xi}}_i - \dot{\boldsymbol{\xi}}_i^*(t-T)\|\|\boldsymbol{\xi}_j^*(t-T)\| + \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^*(t-T)\|\|\dot{\boldsymbol{\xi}}_j^*(t-T)\| \\ &= \|\dot{\boldsymbol{\xi}}_i - \dot{\boldsymbol{\xi}}_i^*(t-T)\| + \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_i^*(t-T)\| \\ &\leq T^{\frac{1}{2}}(A^{aM} + V^{aM}). \end{aligned} \tag{A.13}$$

By using the equality in (A.10), the translational and angular velocity errors are then upper-bounded as

$$\begin{aligned} \|\boldsymbol{e}_i^{lv}\| &\leq \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big) T^{\frac{1}{2}} A^{lM}, && \text{(A.14)} \\ \|\boldsymbol{e}_i^{av}\| &\leq \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big) T^{\frac{1}{2}} (A^{aM} + V^{aM}). && \text{(A.15)} \end{aligned}$$

Reaching the auxiliary switching surface implies that the sliding surface in (4.17) has been reached, i.e., $\boldsymbol{s}_i = \mathbf{0}$, resulting in $\|\boldsymbol{e}_i^p\| = \beta_i \|\boldsymbol{e}_i^v\|^{\alpha_i}$. Consider the velocity error bounds in (A.14) and (A.15), the pose error bounds satisfy the inequalities

$$\|\boldsymbol{e}_i^{lp}\| \leq \beta_i \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{\alpha_i} T^{\frac{\alpha_i}{2}} (A^{lM})^{\alpha_i}, \tag{A.16}$$

$$\|\boldsymbol{e}_i^{ap}\| \leq \beta_i \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{\alpha_i} T^{\frac{\alpha_i}{2}} (A^{aM} + V^{aM})^{\alpha_i}. \tag{A.17}$$

Therefore, (A.16) and (A.17) show that the pose errors are upper-bounded by network delays T, control gains α_i and β_i, and the maximum assigned velocity and acceleration A_i^{lM}, A_i^{aM}, and V_i^{aM}. (A.16) and (A.17) also indicate the effect of the network connectivity, which is not discussed in this paper as the fixed topology is assumed. This concludes the proof. □

A.3 Stability Proof of Theorem 4.3

Proof. In what follows, the tracking stability and convergence proof are provided. Substituting the control input in (4.39) into the dynamics in (3.28) and cancelling out $\boldsymbol{g}_i$ yield

$$\begin{aligned} M_i \ddot{\boldsymbol{x}}_i \;=\; & C_i \Delta_{\dot{\boldsymbol{x}}} + \Big(\sum_{j=1}^{n} \Omega_j + b_i\Big)^{-1} M_i \Big\{ \frac{-(\dot{\boldsymbol{e}}_i)^{(2-\alpha_i)}}{\alpha_i \beta_i} + \sum_{j=1}^{n} \Omega_j \ddot{\hat{\boldsymbol{x}}}_j (t - T_{ij}(t)) \\ & + b_i \ddot{\boldsymbol{x}}_0 (t - T_{i0}(t)) - \Big(\sum_{j=1}^{n} \Omega_j + b_i\Big) C_i \mathbb{L}_i M_i^{M+} \mathrm{sgn}(\boldsymbol{s}_i) - \kappa_i \mathrm{sgn}(\boldsymbol{s}_i) \Big\}. \end{aligned} \tag{A.18}$$

Multiplying both sides by $\Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)M_i^+$ gives

$$\begin{aligned}\Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)\ddot{\boldsymbol{x}}_i &= \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)M_i^+ C_i\Delta_{\dot{\boldsymbol{x}}} - \frac{(\dot{\boldsymbol{e}}_i)^{(2-\alpha_i)}}{\alpha_i\beta_i} + \sum_{j=1}^{n}\Omega_j\ddot{\hat{\boldsymbol{x}}}_j(t-T_{ij}(t)) \\ &\quad + b_i\ddot{\boldsymbol{x}}_0(t-T_{i0}(t)) - \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)C_i\mathbb{L}_i M_i^{M+}\mathrm{sgn}(\boldsymbol{s}_i) - \kappa_i\mathrm{sgn}(\boldsymbol{s}_i).\end{aligned}$$

Using the equation of $\Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)\ddot{\boldsymbol{x}}_i - \sum_{j=1}^{n}\Omega_j\ddot{\hat{\boldsymbol{x}}}_j(t-T_{ij}(t)) - b_i\ddot{\boldsymbol{x}}_0(t-T_{i0}(t)) = \ddot{\boldsymbol{e}}_i$ results in

$$\ddot{\boldsymbol{e}}_i = \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)M_i^+ C_i\Delta_{\dot{\boldsymbol{x}}} - \frac{(\dot{\boldsymbol{e}}_i)^{(2-\alpha_i)}}{\alpha_i\beta_i} - \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)C_i\mathbb{L}_i M_i^{M+}\mathrm{sgn}(\boldsymbol{s}_i) - \kappa_i\mathrm{sgn}(\boldsymbol{s}_i).$$

Multiplying both sides by $\alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})$ gives

$$\begin{aligned}\dot{\boldsymbol{e}}_i + \alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})\ddot{\boldsymbol{e}}_i &= \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)\alpha_i\beta_i M_i^+\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})C_i\Delta_{\dot{\boldsymbol{x}}} \\ &\quad - \Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)\alpha_i\beta_i C_i\mathbb{L}_i M_i^{M+}\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})\mathrm{sgn}(\boldsymbol{s}_i) \\ &\quad - \alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})\kappa_i\mathrm{sgn}(\boldsymbol{s}_i). \qquad \text{(A.19)}\end{aligned}$$

Since the time derivative of the sliding surface is $\dot{\boldsymbol{s}}_i = \dot{\boldsymbol{e}}_i + \alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})\ddot{\boldsymbol{e}}_i$, we have

$$\begin{aligned}\dot{\boldsymbol{s}}_i &= -\Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)\alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})[M_i^{M+}C_i\mathbb{L}_i\mathrm{sgn}(\boldsymbol{s}_i) - M^+C_i\Delta_{\dot{\boldsymbol{x}}}] \\ &\quad - \alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})\kappa_i\mathrm{sgn}(\boldsymbol{s}_i). \qquad \text{(A.20)}\end{aligned}$$

Constructing the Lyapunov function in a quadratic form as

$$V_i = \frac{1}{2}\boldsymbol{s}_i^T\boldsymbol{s}_i.$$

Using (A.20), the derivative of V_i becomes

$$\begin{aligned}\dot{V}_i &= \boldsymbol{s}_i^T\dot{\boldsymbol{s}}_i \\ &= -\Big(\sum_{j=1}^{n}\Omega_j + b_i\Big)\alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})[M_i^{M+}C_i\mathbb{L}_i\|\boldsymbol{s}_i\| - \boldsymbol{s}_i^T M^+C_i\Delta_{\dot{\boldsymbol{x}}}] \\ &\quad - \alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})\kappa_i\|\boldsymbol{s}_i\|. \qquad \text{(A.21)}\end{aligned}$$

Since $\boldsymbol{s}_i^T M^+\Delta_{\dot{\boldsymbol{x}}} \leq \|\boldsymbol{s}_i\|M_i^{M+}\mathbb{L}_i$, (A.21) becomes

$$\dot{V}_i \leq -\alpha_i\beta_i\mathrm{diag}(\dot{\boldsymbol{e}}_i^{\alpha_i-1})\kappa_i\|\boldsymbol{s}_i\| \leq 0.$$

Therefore, the proposed control system is stable and the sliding surface can be reached, i.e., $\boldsymbol{s}_i \to \boldsymbol{0}$. Furthermore, when the sliding surface is reached, we have

$$\dot{\boldsymbol{e}}_i = -\beta_i^{-\frac{1}{\alpha_i}} \boldsymbol{e}_i^{-\frac{1}{\alpha_i}}. \tag{A.22}$$

To prove the motion tracking convergence, another Lyapunov function is defined as $V_i^e = \frac{1}{2}\boldsymbol{e}_i^T \boldsymbol{e}_i$ and its derivative is

$$\dot{V}_i^e = \boldsymbol{e}_i^T \dot{\boldsymbol{e}}_i = -\beta_i^{-\frac{1}{\alpha_i}} \|\boldsymbol{e}_i\|_1^{1-\frac{1}{\alpha_i}}. \tag{A.23}$$

As $\beta_i > 0$ we have $\dot{V}_i^e \leq 0$, which proves that the tracking error can ultimately converge to zero. This concludes the proof. □

Appendix B

Theoratical Proof in Chapter 5

B.1 Proof of Theorem 5.1

Proof. Substituting the control input signals in (5.16) into the dynamics in (5.1) and cancelling out $\bar{C}_i\dot{\boldsymbol{x}}_i + \bar{\boldsymbol{g}}_i$ yield the closed-loop dynamics:

$$\begin{aligned}
\bar{M}_i\ddot{\boldsymbol{x}}_i \;=\;& \boldsymbol{\rho}_i + \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)^{-1}\bar{M}_i\Bigg\{\frac{-(\boldsymbol{e}_i^v)^{(2-\alpha_i)}}{\alpha_i\beta_i} + \sum_{j=1}^{n} a_{ij}\ddot{\hat{\boldsymbol{x}}}_j(t-T_{ij}) \\
& + b_i\ddot{\boldsymbol{x}}_0(t-T_{i0}) - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\bar{M}_i^{M+}\hat{\Theta}_i\Phi_i\mathrm{sgn}(\boldsymbol{s}_i) \\
& - \Big[\mathbb{B}_i^{ae} + \mathbb{B}_i^{tv} + \frac{1}{\alpha_i\beta_i}\mathbb{B}_i^{sd}\mathrm{diag}((\boldsymbol{e}_i^v)^{(1-\alpha_i)})\Big]\mathrm{sgn}(\boldsymbol{s}_i) - \kappa_i\mathrm{sgn}(\boldsymbol{s}_i)\Bigg\}.
\end{aligned}$$

Multiplying $(\sum_{j=1}^n a_{ij} + b_i)\bar{M}_i^+$ at both sides, and moving $\sum_{j=1}^n a_{ij}\ddot{\hat{\boldsymbol{x}}}_j(t-T_{ij}) + b_i\ddot{\boldsymbol{x}}_0(t-T_{i0})$ to the left hand side, the closed-loop dynamics of the i^{th} manipulator becomes

$$\begin{aligned}
& (\sum_{j=1}^{n} a_{ij} + b_i)\ddot{\boldsymbol{x}}_i - \sum_{j=1}^{n} a_{ij}\ddot{\hat{\boldsymbol{x}}}_j(t-T_{ij}) - b_i\ddot{\boldsymbol{x}}_0(t-T_{i0}) \\
=\;& (\sum_{j=1}^{n} a_{ij} + b_i)\bar{M}_i^+\boldsymbol{\rho}_i + \frac{-(\boldsymbol{e}_i^v)^{(2-\alpha_i)}}{\alpha_i\beta_i} - \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\bar{M}_i^{M+}\hat{\Theta}_i\Phi_i\mathrm{sgn}(\boldsymbol{s}_i) \\
& - \Big[\mathbb{B}_i^{ae} + \mathbb{B}_i^{tv} + \frac{1}{\alpha_i\beta_i}\mathbb{B}_i^{sd}\mathrm{diag}((\boldsymbol{e}_i^v)^{(1-\alpha_i)})\Big]\mathrm{sgn}(\boldsymbol{s}_i) - \kappa_i\mathrm{sgn}(\boldsymbol{s}_i). \qquad \text{(B.1)}
\end{aligned}$$

According to (5.11) and (5.13), the left-hand side of (B.1):

$$\Big(\sum_{j=1}^{n} a_{ij} + b_i\Big)\ddot{\boldsymbol{x}}_i - \sum_{j=1}^{n} a_{ij}\ddot{\hat{\boldsymbol{x}}}_j(t-T_{ij}) - b_i\ddot{\boldsymbol{x}}_0(t-T_{i0}) = \dot{\boldsymbol{e}}_i^v + \boldsymbol{\varepsilon}_i^{tv} + \boldsymbol{\varepsilon}_i^{ae}. \qquad \text{(B.2)}$$

For the sake of compactness, denote $E_i^v = \mathrm{diag}[(\boldsymbol{e}_i^v)^{(\alpha_i - 1)}]$. Then multiplying $\alpha_i\beta_i E_i^v$ at both sides of (B.1) and using the relations in (5.5), (5.15) and (B.2) lead to

$$\begin{aligned}\dot{\boldsymbol{s}}_i &= -\alpha_i\beta_i E_i^v(\boldsymbol{\varepsilon}_i^{tv}+\boldsymbol{\varepsilon}_i^{ae}) - \alpha_i\beta_i(\mathbb{B}_i^{ae}+\mathbb{B}_i^{tv})E_i^v\mathrm{sgn}(\boldsymbol{s}_i) - \mathbb{B}_i^{sd}\mathrm{sgn}(\boldsymbol{s}_i) - \boldsymbol{\varepsilon}_i^{sd}\\ &\quad -\Big(\sum_{j=1}^n a_{ij}+b_i\Big)\alpha_i\beta_i\bar{M}_i^{M+}\hat{\Theta}_i\Phi_i E_i^v \mathrm{sgn}(\boldsymbol{s}_i) + \Big(\sum_{j=1}^n a_{ij}+b_i\Big)\alpha_i\beta_i E_i^v\bar{M}_i^+\boldsymbol{\rho}_i\\ &\quad -\alpha_i\beta_i\kappa_i E_i^v\mathrm{sgn}(\boldsymbol{s}_i). \end{aligned}\tag{B.3}$$

Constructing the Lyapunov function in a quadratic form as

$$V_i = \frac{1}{2}\boldsymbol{s}_i^T\boldsymbol{s}_i + \frac{1}{2}\tilde{\Theta}_i^2, \tag{B.4}$$

where $\tilde{\Theta}_i$ is the estimating error of Θ_i, i.e., $\tilde{\Theta}_i = \Theta_i - \hat{\Theta}_i$.

Using (B.3), the derivative of V_i becomes

$$\begin{aligned}\dot{V}_i &= \boldsymbol{s}_i^T\dot{\boldsymbol{s}}_i + \tilde{\Theta}_i\dot{\tilde{\Theta}}_i\\ &= -\alpha_i\beta_i\boldsymbol{s}_i^T E_i^v(\boldsymbol{\varepsilon}_i^{tv}+\boldsymbol{\varepsilon}_i^{ae}) - \alpha_i\beta_i(\mathbb{B}_i^{ae}+\mathbb{B}_i^{tv})\boldsymbol{s}_i^T E_i^v\mathrm{sgn}(\boldsymbol{s}_i) - \mathbb{B}_i^{sd}\boldsymbol{s}_i^T\mathrm{sgn}(\boldsymbol{s}_i) - \boldsymbol{s}_i^T\boldsymbol{\varepsilon}_i^{sd}\\ &\quad -\Big(\sum_{j=1}^n a_{ij}+b_i\Big)\alpha_i\beta_i\bar{M}_i^{M+}\hat{\Theta}_i\Phi_i\boldsymbol{s}_i^T E_i^v\mathrm{sgn}(\boldsymbol{s}_i) + \Big(\sum_{j=1}^n a_{ij}+b_i\Big)\alpha_i\beta_i\boldsymbol{s}_i^T E_i^v\bar{M}_i^+\boldsymbol{\rho}_i\\ &\quad +\tilde{\Theta}_i\dot{\tilde{\Theta}}_i - \alpha_i\beta_i\kappa_i\boldsymbol{s}_i^T E_i^v\mathrm{sgn}(\boldsymbol{s}_i).\end{aligned}$$

As $\boldsymbol{s}_i^T\mathrm{sgn}(\boldsymbol{s}_i) \geq 0$, and according to the definitions of the compensatory bounds, we have the following inequalities: $-\boldsymbol{s}_i^T\boldsymbol{\varepsilon}_i^{sd} - \mathbb{B}_i^{sd}\boldsymbol{s}_i^T\mathrm{sgn}(\boldsymbol{s}_i) \leq 0$, and $-\alpha_i\beta_i\boldsymbol{s}_i^T E_i^v(\boldsymbol{\varepsilon}_i^{tv} + \boldsymbol{\varepsilon}_i^{ae}) - \alpha_i\beta_i(\mathbb{B}_i^{ae}+\mathbb{B}_i^{tv})\boldsymbol{s}_i^T E_i^v\mathrm{sgn}(\boldsymbol{s}_i) \leq 0$, Then, $\dot{V}_i$ becomes

$$\begin{aligned}\dot{V}_i &\leq -\Big(\sum_{j=1}^n a_{ij}+b_i\Big)\alpha_i\beta_i\bar{M}_i^{M+}\hat{\Theta}_i\Phi_i\boldsymbol{s}_i^T E_i^v\mathrm{sgn}(\boldsymbol{s}_i) + \Big(\sum_{j=1}^n a_{ij}+b_i\Big)\alpha_i\beta_i\boldsymbol{s}_i^T E_i^v\bar{M}_i^+\boldsymbol{\rho}_i\\ &\quad +\tilde{\Theta}_i\dot{\tilde{\Theta}}_i - \alpha_i\beta_i\kappa_i\boldsymbol{s}_i^T E_i^v\mathrm{sgn}(\boldsymbol{s}_i)\\ &\leq -\alpha_i\beta_i\Big(\sum_{j=1}^n a_{ij}+b_i\Big)\|\boldsymbol{s}_i^T E_i^v\|_1\bar{M}_i^{M+}(\hat{\Theta}_i\Phi_i - \Theta_i\Phi_i) - \alpha_i\beta_i\kappa_i\|\boldsymbol{s}_i^T E_i^v\|_1 + \tilde{\Theta}_i\dot{\tilde{\Theta}}_i.\end{aligned}$$

Let the adaptive law be

$$\dot{\hat{\Theta}}_i = -\alpha_i\beta_i\Big(\sum_{j=1}^n a_{ij}+b_i\Big)\|\boldsymbol{s}_i^T E_i^v\|_1\bar{M}_i^{M+}\Phi_i, \tag{B.5}$$

and use the fact that $\dot{\hat{\Theta}}_i = \dot{\tilde{\Theta}}_i$, we have $\dot{V}_i$ as

$$\dot{V}_i \leq -\alpha_i\beta_i\kappa_i\|\boldsymbol{s}_i^T E_i^v\|_1. \tag{B.6}$$

When $\boldsymbol{e}_i^v \neq \mathbf{0}$, the finite time stability of the delayed systems can be realized. According to Lemma 3.3, the sliding surface $\boldsymbol{s}_i = \mathbf{0}$ is reachable in a finite time t_i^s that is given by

$$t_i^s \leq \frac{\sqrt{6} V_i(x_0)^{\frac{1}{2}}}{3\alpha_i \beta_i \kappa_i \|(\boldsymbol{e}_i^v)^{(\alpha_i - 1)}\|_1}. \tag{B.7}$$

This concludes the proof. $\square$

B.2 Proof of Theorem 5.2

Proof. When $\boldsymbol{e}_i^v = \mathbf{0}$, $\boldsymbol{s}_i \neq \mathbf{0}$, and $\tilde{\Theta}_i \neq 0$, from (B.1) and (B.2), we have

$$\begin{aligned} \dot{\boldsymbol{e}}_i^v \;=\; & -(\mathbb{B}_i^{ae} + \mathbb{B}_i^{tv})\mathrm{sgn}(\boldsymbol{s}_i) - (\boldsymbol{\varepsilon}_i^{tv} + \boldsymbol{\varepsilon}_i^{ae}) - \kappa_i \mathrm{sgn}(\boldsymbol{s}_i) \\ & + \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big) \bar{M}_i^{M+} (\boldsymbol{\rho}_i - \hat{\Theta}_i \Phi_i \mathrm{sgn}(\boldsymbol{s}_i)). \end{aligned} \tag{B.8}$$

Consider the following properties of the vector norm: (1) $\boldsymbol{x} \in \mathbb{R}^{m\times 1}$ and $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$ $(m = 1, 2, \cdots)$, $\|\boldsymbol{x}\|_1 = 0$ if and only if $\boldsymbol{x} = \mathbf{0}$; (2) $|\|\boldsymbol{x}\|_1 - \|\boldsymbol{y}\|_1| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_1$, we have

$$\|\dot{\boldsymbol{e}}_i^v\|_1 \geq \left| \|\mathbb{B}_i^{ae} + \mathbb{B}_i^{tv}\|_1 - \|\boldsymbol{\varepsilon}_i^{tv} + \boldsymbol{\varepsilon}_i^{ae}\|_1 + \kappa_i + \Big(\sum_{j=1}^{n} a_{ij} + b_i\Big) \bar{M}_i^{M+} |\|\boldsymbol{\rho}_i\| - \hat{\Theta}_i \Phi_i| \right|.$$

From the definitions of the compensatory bounds and adaption bounds, we have $\|\dot{\boldsymbol{e}}_i^v\|_1 > 0$ and therefore $\dot{\boldsymbol{e}}_i^v \neq \mathbf{0}$. This shows that $\boldsymbol{e}_i^v = \mathbf{0}$ is not an attractor for $\boldsymbol{s}_i \neq \mathbf{0}$ and $\tilde{\Theta}_i \neq 0$ [108]. When the sliding surface is reached, $\boldsymbol{s}_i = \mathbf{0}$ and making the time derivative on both sides of (5.15) leads to

$$\dot{\boldsymbol{e}}_i^v = \frac{1}{\alpha_i \beta_i} \mathrm{diag}[(\boldsymbol{e}_i^v)^{1-\alpha_i}](-\boldsymbol{e}_i^v + \boldsymbol{\varepsilon}_i^{sd}). \tag{B.9}$$

Considering the fact that $(\boldsymbol{e}_i^v)^{1-\alpha_i} \geq \mathbf{0}$, (B.9) can be expressed by the element-wise inequalities as $(p = 1, 2, 3)$,

$$\begin{cases} \dot{\boldsymbol{e}}_i^v(p) \leq 0 \text{ when } -\boldsymbol{e}_i^v(p) + \boldsymbol{\varepsilon}_i^{sd}(p) \leq 0, \\ \dot{\boldsymbol{e}}_i^v(p) \geq 0 \text{ when } -\boldsymbol{e}_i^v(p) + \boldsymbol{\varepsilon}_i^{sd}(p) \geq 0. \end{cases} \tag{B.10}$$

The monotonicity in (B.10) implies that there exists a switching surface given by

$$\mathrm{abs}(\boldsymbol{e}_i^v) = \mathrm{abs}(\boldsymbol{\varepsilon}_i^{sd}), \;\; \text{with } \; \mathrm{sgn}(\boldsymbol{e}_i^v) = \mathrm{sgn}(\boldsymbol{\varepsilon}_i^{sd}), \tag{B.11}$$

that can always be reached with any initial velocity error. When the sliding surface in (5.15) and the switching surfaces in (B.11) are all reached, the position error $\boldsymbol{e}_i^p$ can be expressed as

$$\boldsymbol{e}_i^p = -\beta_i\Big(\sum_{j=1}^n a_{ij} + b_i\Big)^{\alpha_i} \mathrm{diag}(|\boldsymbol{\varepsilon}_i^{sd}|^{\alpha_i})\mathrm{sgn}(\boldsymbol{\varepsilon}_i^{sd}),$$

where $| * |$ is the absolute operation.

From (5.7), the position tracking error $\boldsymbol{e}_i^p$ is bounded, that is,

$$\|\boldsymbol{e}_i^p\|_1 \leq \beta_i\Big(\sum_{j=1}^n a_{ij} + b_i\Big)^{\alpha_i} (\mathbb{B}_i^{sd})^{\alpha_i}. \tag{B.12}$$

To determine the convergence time to the switching surface in (B.11), let $\boldsymbol{\varepsilon}_i^t(t) \in \mathbb{R}^{3\times 1}$ denote the distance (along the direction of $\boldsymbol{e}_i^v$) of the velocity error state to the switching surface at time step t, and then the velocity error can be expressed as

$$\boldsymbol{e}_i^v(t) = \boldsymbol{\varepsilon}_i^{sd}(t) + \boldsymbol{\varepsilon}_i^t(t). \tag{B.13}$$

From the monotonicity in (B.10), there always exists an time-varying diagonal matrix $K(t) \in \mathbb{R}^{3\times 3}$ such that $\boldsymbol{\varepsilon}_i^t(t) = K(t)\boldsymbol{e}_i^v(t)$. Substituting (B.13) in (B.9) leads to

$$\begin{aligned} \dot{\boldsymbol{e}}_i^v(t) &= \frac{1}{\alpha_i\beta_i} diag[(\boldsymbol{e}_i^v(t))^{1-\alpha_i}]\boldsymbol{\varepsilon}_i^t(t) = \frac{K(t)}{\alpha_i\beta_i}(\boldsymbol{e}_i^v(t))^{2-\alpha_i} \\ &= -\alpha_i^{-1}\beta_i^{-\frac{2}{\alpha_i}} K(t)[\boldsymbol{e}_i^p(t)]^{\frac{2-\alpha_i}{\alpha_i}}. \end{aligned} \tag{B.14}$$

Assume that when $\boldsymbol{s}_i(t_i^s) = \boldsymbol{0}$ at t_i^s in (B.7), the switching surface in (B.11) has not been reached yet and then the convergence time t_i^e counts from t_i^s, that is,

$$t_i^e = \frac{\|K(t_i^s)\boldsymbol{e}_i^v(t_i^s) - K(t_i^s + t_i^e)\boldsymbol{e}_i^v(t_i^s + t_i^e)\|_1}{\|\dot{\boldsymbol{e}}_i^v(t)\|_1}. \tag{B.15}$$

As $K(t_i^s + t_i^e) = \mathrm{diag}[0 \quad 0 \quad 0]^T$ when the switching surface is reached, substituting (B.14) into (B.15), the convergence time becomes

$$\begin{aligned} t_i^e &= \frac{\|K(t_i^s)\boldsymbol{e}_i^v(t_i^s)\|_1}{\|K(t)\|_1\alpha_i^{-1}\beta_i^{-\frac{2}{\alpha_i}}\Big\|\boldsymbol{e}_i^p(t)\Big\|_1^{\frac{2-\alpha_i}{\alpha_i}}} \\ &\leq \alpha_i\beta_i^{\frac{2}{\alpha_i}}\|K(t)\|_1^{-1}\|K(t_i^s)\|_1\|\boldsymbol{e}_i^v(t_i^s)\|_1\Big\|\boldsymbol{e}_i^{\mathrm{pmin}}\Big\|_1^{1-\frac{2}{\alpha_i}}. \end{aligned}$$

where $\boldsymbol{e}_i^{\mathrm{pmin}}$ is the minimal of $\boldsymbol{e}_i^p(t)$, which does always exist. $\|K(t)\|_1$ is bounded considering the fact of the monotonicity in (B.10). Therefore, t_i^e is a finite time, and the length is related to the control gains α_i and β_i. This concludes the proof. □

Appendix C

Theoratical Proof in Chapter 6

C.1 Proof of Theorem 6.1

Proof. The closed-loop dynamics of the master manipulator are

$$\begin{aligned}\bar{M}_m\ddot{\boldsymbol{x}}_m+\bar{C}_m\dot{\boldsymbol{x}}_m+\bar{\boldsymbol{g}}_m &= \bar{C}_m\dot{\boldsymbol{x}}_m+\bar{\boldsymbol{g}}_m-\boldsymbol{f}_h+\bar{M}_m\Big\{\frac{-(\boldsymbol{e}_m^v)^{(2-\alpha_m)}}{\alpha_m\beta_m}+\ddot{\boldsymbol{x}}_r\\ &\quad -\bar{M}_m^{M+}\hat{\Theta}_m\Phi_m\mathrm{sgn}(\boldsymbol{s}_m)-\kappa_m\mathrm{sgn}(\boldsymbol{s}_m)\Big\}+\boldsymbol{\rho}_m+\boldsymbol{f}_h.\end{aligned}$$

Cancelling out common terms yields

$$\ddot{\boldsymbol{x}}_m=\frac{-(\boldsymbol{e}_m^v)^{(2-\alpha_m)}}{\alpha_m\beta_m}+\ddot{\boldsymbol{x}}_r-\bar{M}_m^{M+}\hat{\Theta}_m\Phi_m\mathrm{sgn}(\boldsymbol{s}_m)-\kappa_m\mathrm{sgn}(\boldsymbol{s}_m)+\bar{M}_m^{+}\boldsymbol{\rho}_m.$$

From (6.13), we have $\dot{\boldsymbol{e}}_m^v=\ddot{\boldsymbol{x}}_m-\ddot{\boldsymbol{x}}_r$ and multiplying $\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}$ on both sides and using the relation $\dot{\boldsymbol{s}}_m=\boldsymbol{e}_m^v+\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\dot{\boldsymbol{e}}_m^v$, we obtain

$$\begin{aligned}\dot{\boldsymbol{s}}_m &= \boldsymbol{e}_m^v+\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\dot{\boldsymbol{e}}_m^v\\ &= -\bar{M}_m^{M+}\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\hat{\Theta}_m\Phi_m\mathrm{sgn}(\boldsymbol{s}_m)+\bar{M}_m^{+}\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\boldsymbol{\rho}_m\\ &\quad -\kappa_m\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\mathrm{sgn}(\boldsymbol{s}_m).\end{aligned}$$

A continuously differentiable, positive definite candidate Lyapunov function, V_m, is defined as

$$V_m=\frac{1}{2}\boldsymbol{s}_m^T\boldsymbol{s}_m+\frac{1}{2}\tilde{\Theta}_m^2. \tag{C.1}$$

The time derivative of V_m gives

$$\begin{aligned}\dot{V}_m &= \boldsymbol{s}_m^T\dot{\boldsymbol{s}}_m+\tilde{\Theta}_m\dot{\tilde{\Theta}}_m\\ &= -\bar{M}_m^{M+}\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\hat{\Theta}_m\Phi_m\boldsymbol{s}_m^T\mathrm{sgn}(\boldsymbol{s}_m)+\bar{M}_m^{+}\alpha_m\beta_m\boldsymbol{s}_m^T(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\boldsymbol{\rho}_m\\ &\quad -\kappa_m\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\boldsymbol{s}_m^T\mathrm{sgn}(\boldsymbol{s}_m)+\tilde{\Theta}_m\dot{\tilde{\Theta}}_m.\end{aligned} \tag{C.2}$$

Consider the inequalities of $\bar{M}_m^{M+}\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)} \geq 0$, $\boldsymbol{s}_m^T\text{sgn}(\boldsymbol{s}_m) \geq 0$, and $\boldsymbol{\rho}_m \leq \Theta_m\Phi_m$, (C.2) becomes

$$\begin{aligned}\dot{V}_m \quad &\leq \quad \bar{M}_m^{M+}\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\tilde{\Theta}_m\Phi_m\|\boldsymbol{s}_m^T\| + \tilde{\Theta}_m\dot{\tilde{\Theta}}_m \\ &\quad -\kappa_m\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\boldsymbol{s}_m^T\text{sgn}(\boldsymbol{s}_m).\end{aligned}$$

Substituting the adaptive law in (6.12), we get

$$\dot{V}_m \leq -\kappa_m\alpha_m\beta_m(\boldsymbol{e}_m^v)^{(\alpha_m-1)}\boldsymbol{s}_m^T\text{sgn}(\boldsymbol{s}_m) \leq 0.$$

According to Lemma 3.3, when $\boldsymbol{e}_m^v \neq \mathbf{0}$, $\boldsymbol{s}_m = \mathbf{0}$ and the stability of the slave control system can be realized in a finite time. As well, follow the procedure in (A.22)-(A.23), it can be proved that $\boldsymbol{e}_m^p \to \mathbf{0}$ when $\boldsymbol{s}_m = \mathbf{0}$. $\square$

C.2 Proof of Theorem 6.2

In the proof of Theorem 6.2, a Lyapunov-based stability analysis is provided to conclude the finite-time convergence of sliding surface.

Proof. Substituting the control input (6.18) into the dynamics in (6.2) yields the closed-loop dynamics:

$$\begin{aligned}\bar{M}_s\ddot{\boldsymbol{x}}_s + \bar{C}_s\dot{\boldsymbol{x}}_s + \bar{\boldsymbol{g}}_s \quad &= \quad \bar{C}_s\dot{\boldsymbol{x}}_s + \bar{\boldsymbol{g}}_s + \bar{M}_s\Big\{\frac{-(\boldsymbol{e}_s^v)^{(2-\alpha_s)}}{\alpha_s\beta_s} + \ddot{\boldsymbol{x}}_m(t-T_1(t)) \\ &\quad -\Big(\frac{1}{\alpha_s\beta_s}\mathbb{B}_s^{sd}\text{diag}(\boldsymbol{e}_s^v)^{(1-\alpha_s)} + \mathbb{B}_s^a\Big)\text{sgn}(\boldsymbol{s}_s) \\ &\quad -\bar{M}_s^{M+}\hat{\Theta}_s\Phi_s\text{sgn}(\boldsymbol{s}_s) - \kappa_s\text{sgn}(\boldsymbol{s}_s)\Big\} + \boldsymbol{f}_e + \boldsymbol{\rho}_s - \boldsymbol{f}_e. \quad \text{(C.3)}\end{aligned}$$

Then it follows that

$$\begin{aligned}\ddot{\boldsymbol{x}}_s - \ddot{\boldsymbol{x}}_m(t-T_1(t)) \quad &= \quad \frac{-(\boldsymbol{e}_s^v)^{(2-\alpha_s)}}{\alpha_s\beta_s} - \Big(\frac{1}{\alpha_s\beta_s}\mathbb{B}_s^{sd}\text{diag}(\boldsymbol{e}_s^v)^{(1-\alpha_s)} + \mathbb{B}_s^a\Big)\text{sgn}(\boldsymbol{s}_s) \\ &\quad -\bar{M}_s^{M+}\hat{\Theta}_s\Phi_s\text{sgn}(\boldsymbol{s}_s) - \kappa_s\text{sgn}(\boldsymbol{s}_s) + \bar{M}_i^+\boldsymbol{\rho}_s. \quad \text{(C.4)}\end{aligned}$$

The left-hand-side of (C.4) has the relationship

$$\ddot{\boldsymbol{x}}_s - \ddot{\boldsymbol{x}}_m(t-T_1(t)) \quad = \quad \dot{\boldsymbol{e}}_s^v + \boldsymbol{\varepsilon}_s^a, \quad \text{(C.5)}$$

where $\boldsymbol{\varepsilon}_s^a = \ddot{\boldsymbol{x}}_m(t - T_1(t))(1 - \dot{\delta}_1(t))$ that is upper-bounded by $\mathbb{B}_s^a(t)$ given in (6.16). The relation in (C.5) facilitates the expression of time-derivative of sliding variable $\dot{\boldsymbol{s}}_s$

$$\begin{aligned}\dot{\boldsymbol{s}}_s &= \alpha_s\beta_s\mathrm{diag}[(\boldsymbol{e}_s^v)^{(\alpha_s-1)}]\Big[\boldsymbol{\varepsilon}_s^a - \mathbb{B}_s^a\mathrm{sgn}(\boldsymbol{s}_s) + \bar{M}_i^+\boldsymbol{\rho}_s - \bar{M}_s^{M+}\hat{\Theta}_s\Phi_s\mathrm{sgn}(\boldsymbol{s}_s)\\ &\quad -\kappa_s(t)\mathrm{sgn}(\boldsymbol{s}_s)\Big] + \boldsymbol{\varepsilon}_s^{sd} - \mathbb{B}_s^{sd}\mathrm{sgn}(\boldsymbol{s}_s). \qquad (C.6)\end{aligned}$$

A continuously differentiable, positive definite candidate Lyapunov function is defined as

$$V_s = \frac{1}{2}\boldsymbol{s}_s^T\boldsymbol{s}_s + \frac{1}{2}\tilde{\Theta}_s^2, \qquad (C.7)$$

where $\tilde{\Theta}_s$ is the estimating error of Θ_s, i.e., $\tilde{\Theta}_s = \Theta_s - \hat{\Theta}_s$, and $\dot{\hat{\Theta}}_s = \dot{\tilde{\Theta}}_s$. Taking derivative of V_s and using (C.6), we get

$$\begin{aligned}\dot{V}_s &= \boldsymbol{s}_s^T\dot{\boldsymbol{s}}_s + \tilde{\Theta}_s\dot{\tilde{\Theta}}_s\\ &= \tilde{\Theta}_i\dot{\tilde{\Theta}}_i + \alpha_s\beta_s\mathrm{diag}[(\boldsymbol{e}_s^v)^{(\alpha_s-1)}]\Big[\boldsymbol{s}_s^T\boldsymbol{\varepsilon}_s^a - \mathbb{B}_s^a\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s) + \bar{M}_i^+\boldsymbol{s}_s^T\boldsymbol{\rho}_s\\ &\quad -\bar{M}_s^{M+}\hat{\Theta}_s\Phi_s\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s) - \kappa_s(t)\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s)\Big] + \boldsymbol{s}_s^T\boldsymbol{\varepsilon}_s^{sd} - \mathbb{B}_s^{sd}\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s). \quad (C.8)\end{aligned}$$

Exploiting the fact that $\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s) \geq 0$, we obtain the following inequalities:

$$\begin{aligned}&\boldsymbol{s}_s^T\boldsymbol{\varepsilon}_s^{sd} - \mathbb{B}_s^{sd}\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s) \leq 0,\\ &\boldsymbol{s}_s^T\boldsymbol{\varepsilon}_s^a - \mathbb{B}_i^a\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s) \leq 0.\end{aligned}$$

Then, $\dot{V}_s$ in (C.8) becomes

$$\begin{aligned}\dot{V}_s &\leq \tilde{\Theta}_s\dot{\tilde{\Theta}}_s + \alpha_s\beta_s\mathrm{diag}[(\boldsymbol{e}_s^v)^{(\alpha_s-1)}]\Big[\|\bar{M}_s^+\|_1\boldsymbol{s}_s^T\Theta_s\Phi_s\\ &\quad -\bar{M}_s^{M+}\hat{\Theta}_s\Phi_s\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s) - \kappa_s(t)\boldsymbol{s}_s^T\mathrm{sgn}(\boldsymbol{s}_s)\Big].\end{aligned}$$

Recall the adaptive law in (6.19), the following inequality can be derived

$$\dot{V}_s \leq -\alpha_s\beta_s\kappa_s\|\boldsymbol{s}_s\|_1\mathrm{diag}[(\boldsymbol{e}_s^v)^{(\alpha_s-1)}] \leq 0. \qquad (C.9)$$

According to Lemma 3.3, when $\boldsymbol{e}_s^v \neq \boldsymbol{0}$, the finite time stability of the slave control system can be realized and the sliding surface $\boldsymbol{s}_s = \boldsymbol{0}$ can be achieved in a finite time t_s given by

$$t_s \leq \frac{\sqrt{6}V_s(x_s(0))^{\frac{1}{2}}}{3\alpha_s\beta_s\kappa_s\|(\boldsymbol{e}_s^v)^{(\alpha_s-1)}\|_1}. \qquad (C.10)$$

Furthermore, the conclusion in (C.10) still holds if it can be proved that $\boldsymbol{e}_s^v = \mathbf{0}$ is not an attractor for $\boldsymbol{s}_s \neq \mathbf{0}$ and $\tilde{\Theta}_s \neq 0$ [108]. Provided that $\boldsymbol{s}_s \neq \mathbf{0}$ and $\tilde{\Theta}_i \neq 0$ when $\boldsymbol{e}_s^v = \mathbf{0}$, the following expression can be obtained from (C.4) and (C.5):

$$\dot{\boldsymbol{e}}_s^v = \bar{M}_i^{+}\boldsymbol{\rho}_s - \boldsymbol{\varepsilon}_s^a - (\bar{M}_s^{M+}\hat{\Theta}_s\Phi_s + \mathbb{B}_s^a + \kappa_s(t))\text{sgn}(\boldsymbol{s}_s). \tag{C.11}$$

To investigate the boundedness of $\dot{\boldsymbol{e}}_s^v$, two basic properties of vector norm are introduced as follows ($\boldsymbol{x} \in \mathbb{R}^{m\times 1}$, $\boldsymbol{y} \in \mathbb{R}^{m\times 1}$, and $m = 1, 2, \cdots$).

- $\|\boldsymbol{x}\|_1 = 0$ if and only if $\boldsymbol{x} = \mathbf{0}$;
- $|\|\boldsymbol{x}\|_1 - \|\boldsymbol{y}\|_1| \leq \|\boldsymbol{x} - \boldsymbol{y}\|_1$.

As a result, (C.11) has the following inequality:

$$\begin{aligned} \|\dot{\boldsymbol{e}}_s^v\|_1 \quad \geq \quad & \Big| \|\bar{M}_s^{+}\boldsymbol{\rho}_s - \bar{M}_s^{M+}\hat{\Theta}_s\Phi_s\text{sgn}(\boldsymbol{s}_s)\|_1 \\ & -\|\kappa_s(t))\text{sgn}(\boldsymbol{s}_s)\|_1 - \|\boldsymbol{\varepsilon}_s^a + \mathbb{B}_s^a\text{sgn}(\boldsymbol{s}_s)\|_1 \Big| > 0. \end{aligned}$$

Therefore, $\|\dot{\boldsymbol{e}}_s^v\|_1 > 0$ implies that $\dot{\boldsymbol{e}}_s^v \neq \mathbf{0}$, which further verifies that $\boldsymbol{e}_s^v = \mathbf{0}$ is not an attractor for $\boldsymbol{s}_s \neq \mathbf{0}$ and $\tilde{\Theta}_s \neq 0$. This concludes the proof of Theorem 6.2. □

C.3 Proof of Theorem 6.3

Proof. Let $\bar{\boldsymbol{e}}_s^p \in \mathbb{R}^3$ denote the position feedback without self-delay as

$$\begin{aligned} \bar{\boldsymbol{e}}_s^p &= \boldsymbol{x}_s(t) - \boldsymbol{x}_m(t - T_1(t)) \\ &= \boldsymbol{x}_s(t) - \boldsymbol{x}_m(t) + \boldsymbol{x}_m(t) - \boldsymbol{x}_m(t - T_1(t)) \\ &= \bar{\boldsymbol{\varepsilon}}_{sm}(t) + [\boldsymbol{x}_m(t) - \boldsymbol{x}_m(t - T_1(t))], \end{aligned} \tag{C.12}$$

where $\bar{\boldsymbol{\varepsilon}}_{sm}(t) = \boldsymbol{x}_s(t) - \boldsymbol{x}_m(t)$ is the tracking error at the current time step between the master and the slave manipulators. When $\bar{\boldsymbol{e}}_s^p \to \mathbf{0}$ [88, 108], using the Euler's approximation of $T_1(t)\dot{\boldsymbol{x}}_m(t) = \boldsymbol{x}_m(t) - \boldsymbol{x}_m(t - T_1(t))$, we have

$$\|\bar{\boldsymbol{\varepsilon}}_{sm}(t)\|_1 = T_1(t)\|\dot{\boldsymbol{x}}_m(t)\|_1. \tag{C.13}$$

On the other hand, the inexactly self-delayed position error is given by

$$\begin{aligned} \boldsymbol{e}_s^p &= \boldsymbol{x}_s(t-\bar{T}) - \boldsymbol{x}_m(t-T_1(t)) \\ &= \boldsymbol{x}_s(t-\bar{T}) - \boldsymbol{x}_m(t-\bar{T}) + \boldsymbol{x}_m(t-\bar{T}) - \boldsymbol{x}_m(t-T_1(t)) \\ &= \boldsymbol{\varepsilon}_{sm}(t-\bar{T}) + [\boldsymbol{x}_m(t-\bar{T}) - \boldsymbol{x}_m(t-T_1(t))] \\ &= \boldsymbol{\varepsilon}_{sm}(t-\bar{T}) + \delta_1(t)\dot{\boldsymbol{x}}_m(t-T_1(t)). \end{aligned} \tag{C.14}$$

Provided that $\bar{\boldsymbol{e}}_s^p \to \boldsymbol{0}$ is realized, from (6.15) and the subsequent result in (C.22), we obtain that $\|\boldsymbol{e}_s^v\|_1 = \|\boldsymbol{\varepsilon}_s^{sd}\|_1 \leq 4\mathbb{B}_v$. Then, $\boldsymbol{\varepsilon}_{sm}(t-\bar{T})$ is upper-bounded by

$$\|\boldsymbol{\varepsilon}_{sm}(t-\bar{T})\|_1 \leq \beta_s(4\mathbb{B}_v)^{\alpha_s} + \delta_1(t)\|\dot{\boldsymbol{x}}_m(t-T_1(t))\|_1. \tag{C.15}$$

In addition, throughout the entire manipulation, we can assume the following equivalent expression of the inequality in (C.15):

$$\|\boldsymbol{\varepsilon}_{sm}(t)\|_1 \leq \beta_s(4\mathbb{B}_v)^{\alpha_s} + \delta_1(t)\|\dot{\boldsymbol{x}}_m(t)\|_1. \tag{C.16}$$

Let an auxiliary error E_s^p define the difference between the the upper bounds of $\bar{\boldsymbol{\varepsilon}}_{sm}(t)$ and $\boldsymbol{\varepsilon}_{sm}(t)$, we obtain

$$E_s^p = \|\boldsymbol{\varepsilon}_{sm}(t)\|_1 - \|\bar{\boldsymbol{\varepsilon}}_{sm}(t)\|_1 \leq \beta_s(4\mathbb{B}_v)^{\alpha_s} - \bar{T}\|\dot{\boldsymbol{x}}_m(t)\|_1. \tag{C.17}$$

In words, the inequality in (C.17) can be interpreted as follows:

1) when $E_s^p > 0$, using the mixed-type feedback signals results in larger tracking errors than using the feedback signals without self-delays.

2) when $E_s^p = 0$, using the mixed-type feedback signals obtains the same tracking accuracy as using the feedback signals without self-delays.

3) when $E_s^p < 0$, controllers using the mixed-type feedback signals provides smaller tracking errors than those using the feedback signals without self-delays.

Therefore, for the ANTSM control system in the presence of network delays, using the mixed-type feedback can improve the tracking accuracy if the control gains satisfy the following sufficient condition:

$$(4^{\alpha_s}\beta_s)^{\frac{1}{1-\alpha_s}} < \mathbb{B}_v\bar{T}^{\frac{1}{1-\alpha_s}}. \tag{C.18}$$

such that $E_s^p < 0$ holds. This concludes the proof. □

C.4 Proof of Proposition 6.1

Proof. When the sliding manifold $\boldsymbol{s}_s = \boldsymbol{0}$ is reached, from (6.17) we have

$$\dot{\boldsymbol{e}}_s^v = \frac{1}{\alpha_s \beta_s} \mathrm{diag}[(\boldsymbol{e}_s^v)^{1-\alpha_s}](-\boldsymbol{e}_s^v + \boldsymbol{\varepsilon}_s^{sd}). \tag{C.19}$$

Since the definition of α_s guarantees that $(\boldsymbol{e}_i^v)^{1-\alpha_s} \geq \boldsymbol{0}$ always holds. (C.19) can be expressed element-wise as ($p = 1, 2, 3$),

$$\begin{cases} \dot{\boldsymbol{e}}_s^v(p) \leq 0, \text{ when } -\boldsymbol{e}_s^v(p) + \boldsymbol{\varepsilon}_s^{sd}(p) \leq 0, \\ \dot{\boldsymbol{e}}_s^v(p) \geq 0, \text{ when } -\boldsymbol{e}_s^v(p) + \boldsymbol{\varepsilon}_s^{sd}(p) \geq 0. \end{cases} \tag{C.20}$$

The monotonicity in (C.20) implies the reachability of a switching surface given by

$$|\boldsymbol{e}_s^v| = |\boldsymbol{\varepsilon}_s^{sd}|, \text{ s.t. } \mathrm{sgn}(\boldsymbol{e}_s^v) = \mathrm{sgn}(\boldsymbol{\varepsilon}_s^{sd}). \tag{C.21}$$

As the switching surface in (C.21) is reached, using (6.13) and (6.14), we obtain

$$\boldsymbol{e}_s^v - \boldsymbol{\varepsilon}_s^{sd} = \dot{\boldsymbol{x}}_s(t - \bar{T}) - \dot{\boldsymbol{x}}_m(t - T_1(t))\dot{\delta}_1 \to \boldsymbol{0}. \tag{C.22}$$

In other words, the delayed master and slave positions are converged and bounded, that is, $\|\dot{\boldsymbol{x}}_s(t-\bar{T})\|_1 \to \|\dot{\boldsymbol{x}}_m(t-T_1(t))\|_1 \leq \mathbb{B}_v$, where $\mathbb{B}_v$ denotes the velocity bound. This concludes the proof. □

C.5 Proof of Proposition 6.2

Proof. Provided that $\dot{\boldsymbol{x}}_m(t) = 0$ and consider (C.22), we obtain $\dot{\boldsymbol{x}}_s(t) = 0$ and $\mathbb{B}_v = 0$ when the sliding surface $\boldsymbol{s}_s = \boldsymbol{0}$ is reached. In other words, the tracking error between the master and slave converges to zero in the sense of $\boldsymbol{\varepsilon}_{sm}(t) = \boldsymbol{x}_s(t) - \boldsymbol{x}_m(t) = 0$. This concludes the proof. □

Appendix D

Transition Waveform

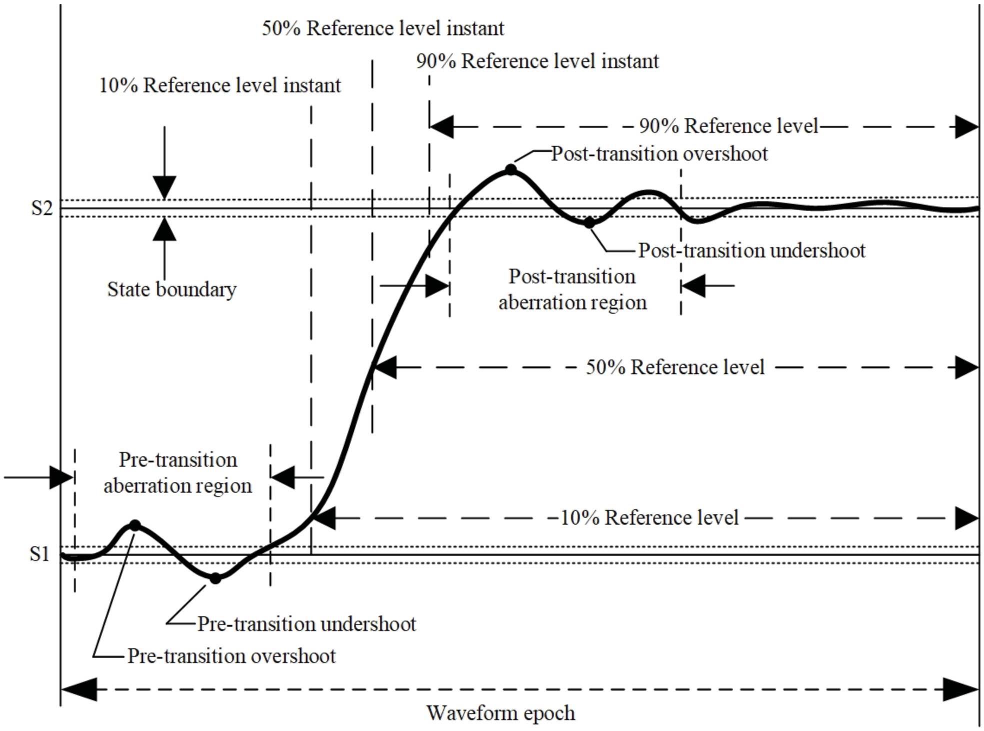


Figure D.1: The positive-going transition waveform [115].

This section briefly provides some basic waveform-related terms and parameters to facilitate the description and discussion of the manipulation performance in this paper (See [115, 116] for a complete description of waveform standards).

Fig.D.1 illustrates the relevant definitions of a two-state (designated by s_1 and s_2, $s_1 < s_2$) waveform with a *transition*. Each state has an associated *State Level*: $level(s_1)$ and $level(s_2)$.

Percent Reference Level

Reference levels are user-specified and constant throughout a waveform epoch. Reference level is usually expressed as a *Percent Reference Level (PRL)* to facilitate the descriptions of other terms and parameters. Commonly used reference levels are 0%, 10%, 50%, 90%, and 100%, which are specified by

$$y_{x\%} = y_{0\%} + \frac{x}{100}(y_{100\%} - y_{0\%}), \tag{D.1}$$

where $x = \{0, 10, 50, 90, 100\}$. $y_{0\%}$ and $y_{100\%}$ are the levels of low state and high state, respectively, e.g., in Fig.D.1, $y_{0\%} = level(s_1)$ and $y_{100\%} = level(s_2)$.

Pre-Transition and Post-Transition Aberration Regions

Aberration Region is the interval between a user-specified instant and a fixed instant. For the *Pre-(Post-)Transition Aberration Region*, the fixed instant is defined as the first sampling instant preceding(succeeding) the 50% *Percent Reference Level (PRL)* instant when the waveform value is within the *State Boundaries* of the state preceding(succeeding) the 50% PRL instant. PRL is user-specified and constant throughout a waveform epoch. The user-specified instant occurs before (after) the fixed instant and is typically equal to the fixed instant minus (plus) three times the transition duration, that is, $t_{post} \mp 3t_{10\%-90\%}$ (or as determined by the user). *State Boundaries* is user-defined and in this paper it is specified as $s_{1,2} \pm 5\% s_{1,2}$.

Overshoot and Undershoot

A single-transition waveform may exhibit four of special aberrations: pre-transition overshoot, pre-transition undershoot, post-transition overshoot and post-transition undershoot, which are determined by

$$OS_{region}(\%) = \frac{y_{max,region} - level(s_k)}{|A|} 100\%, \tag{D.2}$$

$$US_{region}(\%) = \frac{level(s_k) - y_{min,region}}{|A|} 100\%, \tag{D.3}$$

where $region = \{pre, post\}$. A is the *Waveform Amplitude* given by $|A| = |level(s_1) - level(s_2)|$.

Compound Waveform

A compound waveform is completely represented by m states and n transitions where $m + n \geq 4$.

Appendix E

Author's Publications

Journal Papers

1. **H.H. Shen** and Y.J. Pan, "Nonlinear State Estimation and Online Neighbor Selection for Multi-Manipulator Systems", IEEE Transactions on Cybernetics, Revised.

2. U. Ahmad, Y.J. Pan and **H.H. Shen**, "Robust Control Design for Teleoperation of Multiple Mobile Manipulators under Time Delays", International Journal of Robust and Nonlinear Control, Vol.30, No.16, pp.6454-6472, 2020.

3. **H.H. Shen** and Y.J. Pan, "Tracking Synchronization Improvement of Networked Manipulators Using Novel Adaptive Non-Singular Terminal Sliding Mode Control", IEEE Transactions on Industrial Electronics, In Press, March 2020.

4. **H.H. Shen**, Y.J. Pan, U. Ahmad and B.W. He, "Pose Synchronization of Multiple Networked Manipulators using Non-singular Terminal Sliding Mode Control", IEEE/ASME Transactions on Systems, Man and Cybernetics: Systems, In Press, January 2020.

5. **H.H. Shen** and Y.J. Pan, "Improving Tracking Performance of Nonlinear Uncertain Bilateral Teleoperation Systems with Time-Varying Delays and Disturbances", IEEE/ASME Transactions on Mechatronics, Vol.25, No.03, pp.1171-1181, 2020.

Conference Papers

1. R. Adderson, Y.J. Pan and **H.H. Shen**, "Sliding Mode Control for the Formation of a Team of Quadcopters and Mobile Robots", In Proceedings of IEEE International Conference on Mechatronics (ICM2021), March 2021, Kashiwa, Japan, Submitted.

2. **H.H. Shen**, Y.J. Pan and G. Bauer, "Online Noise-Estimation-based Neighbor Selection for Multi-Manipulator Systems", In Proceedings of the 21st IFAC World Congress, July 2020, Berlin, Germany.
3. G. Bauer, Y.J. Pan and **H.H. Shen**, "Adaptive Impedance Control in Bilateral Telerehabilitation with Robotic Exoskeletons", In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2020), October 2020, Toronto, Canada.
4. **H.H. Shen** and Y.J. Pan, "Adaptive Robust Control of Networked Multi-Manipulators with Time-Varying Delays", In Proceedings of the 2019 IEEE American Control Conference, July 2019, Philadelphia, USA, pp. 3670-3675.
5. **H.H. Shen**, Y.J. Pan, U. Ahmad, S. Liu, M, Wu and Y. He, "Tracking Performance Evaluations on the Robust Teleoperative Control of Multiple Manipulators", In Proceedings of the 28th IEEE International Symposium on Industrial Electronics (IEEE-ISIE 2019), June 2019, Vancouver, Canada, pp.1268-1273.
6. **H.H. Shen**, Y.J. Pan and G. Bauer, "Manipulability-based Load Application and Kinematic Decoupling in Cooperative Manipulations", In Proceedings of the 28th IEEE International Symposium on Industrial Electronics (IEEE-ISIE 2019), June 2019, Vancouver, Canada, pp.1168-1173.
7. U. Ahmad, Y.J. Pan, **H.H. Shen** and S. Liu, "Cooperative Control of Mobile Manipulators Transporting an Object based on an Adaptive Backstepping Approach", In Proceedings of the 14th IEEE International Conference on Control and Automation, June 2018, Anchorage, Alaska, USA, pp.198-203.
8. **H.H. Shen**, Y.J. Pan and B. He, "Teleoperation of Multiple Cooperative Slave Manipulators Using Graph-based Non-singular Terminal Sliding-Mode Control", In Proceedings of the IEEE International Conference on Robotics and Biomimetics, December 2017, Macau, China, pp.1430-1435.