# BACKSTEPPING-ADAPTIVE CONTROL OF MOBILE MANIPULATORS FOR TRAJECTORY TRACKING

by

Bhavik Patel

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
August 2016

*To my parents, Arvind and Manisha Patel, Without whom none of my success would be possible.*

# Table of Contents

**List of Tables** . . . . . **v**

**List of Figures** . . . . . **viii**

**Abstract** . . . . . **ix**

**List of Nomenclature** . . . . . **x**

**Acknowledgements** . . . . . **xiii**

**Chapter 1 Introduction** . . . . . **1**

1.1 Background . . . . . 1

1.2 Literature Review . . . . . 2

1.3 Thesis Contribution . . . . . 8

1.4 Thesis Organization . . . . . 9

**Chapter 2 Modeling of Mobile Manipulators** . . . . . **10**

2.1 Kinematics Fundamentals . . . . . 10
2.1.1 Kinematics . . . . . 10
2.1.2 Forward Kinematics and Inverse Kinematics . . . . . 12
2.1.3 Differential Kinematics and the Jacobians . . . . . 15

2.2 System Description . . . . . 16
2.2.1 Kinematic Constraint . . . . . 16
2.2.2 Kinematic Modeling of Mobile Manipulator . . . . . 19
2.2.3 Dynamics of Mobile Manipulators . . . . . 20

**Chapter 3 Proposed Control Design** . . . . . **26**

3.1 Problem Formulation . . . . . 26
3.1.1 Challenging Problems . . . . . 26
3.1.2 Control Objective . . . . . 27
3.1.3 Assumptions . . . . . 27
3.1.4 Velocity Tracking Problem . . . . . 29
3.1.5 Torque Control Problem . . . . . 29

3.2 Control Design . . . . . 30
3.2.1 Kinematic Controller . . . . . 30

3.2.2 Adaptive Torque Control . . . . . 32

**Chapter 4 Simulation Results on Model Based Control . . . . . 35**

4.1 Model Based Control (MBC) . . . . . 36

4.2 Simulation Results . . . . . 39

**Chapter 5 Simulation Results on Backstepping Adaptive Control 43**

5.1 System Model and Control Scheme in MATLAB/Simulink . . . . . 43

5.2 Desired Trajectory Tracking of the End-Effector Position and Vehicle Velocity . . . . . 46
5.2.1 Trajectory Tracking Case 1 . . . . . 47
5.2.2 Trajectory Tracking Case 2 . . . . . 53
5.2.3 Trajectory Tracking Case 3 . . . . . 59

**Chapter 6 Conclusions and Future Work . . . . . 73**

6.1 Conclusions . . . . . 73

6.2 Future Work . . . . . 74

**Bibliography . . . . . 75**

**Appendix A Mathematical Equations . . . . . 79**

A.1 Kinematic Modeling of 5-DOF Mobile Manipulator . . . . . 79

A.2 The Dynamics Model . . . . . 80

A.3 Regressor Matrix . . . . . 84
A.3.1 Regressor Matrix Y in Terms of Vehicle Linear and Angular Velocities . . . . . 87

A.4 MBC control Dynamics . . . . . 89

A.5 MATLAB Codes . . . . . 91

# List of Tables


4.1 Mobile manipulator parameters used in simulation for the MBC control [4]. . . . . 38

4.2 Control gains used for the model based control . . . . . 38

5.1 Mobile manipulator variables . . . . . 44

5.2 Mobile manipulator parameters used in simulation for proposed control . . . . . 44

5.3 Initial position of the end-effector of the mobile manipulator for case 1 . . . . . 48

5.4 Control gain values for case 1 . . . . . 48

5.5 Initial position of the end-effector of the mobile manipulator for Case 2 . . . . . 54

5.6 Control gain values for Case 2 . . . . . 54

5.7 Control gain values for Case 3 . . . . . 61

## List of Figures


2.1 Position and orientation of a rigid body . . . . . 11
2.2 Forward and inverse kinematics [1] . . . . . 12
2.3 PUMA robotic arm [2] . . . . . 13
2.4 Classic D-H Parameters [3] . . . . . 14
2.5 (n+m)-DOF mobile manipulator . . . . . 17
2.6 Kinematic systems . . . . . 18
2.7 5-DOF mobile manipulator system . . . . . 19

3.1 Block diagram of control scheme . . . . . 30

4.1 Simulation procedure in MATLAB/Simulink . . . . . 35
4.2 Simulation block diagram of model based controller . . . . . 36
4.3 Mobile manipulator system considered for the model based control [4] . . . . . 37
4.4 Generalized block diagram of model based control . . . . . 39
4.5 Individual joint position tracking by model-based control. . . . 40
4.6 Joint position trajectory tracking by model-based control. . . . 40
4.7 Trajectory tracking error by model-based control. . . . . 41
4.8 Torque of the joints by model-based control. . . . . 41
4.9 Velocities of the joints by model based control . . . . . 42

5.1 Mobile manipulator considered in the simulation [5] . . . . . 43
5.2 Backstepping-Adaptive Control Scheme . . . . . 45
5.3 Simulink block diagram . . . . . 45
5.4 Desired end-effector trajectories for Case 1 . . . . . 47
5.5 Actual $x_{EE}$ of Case 1 . . . . . 48
5.6 Actual $y_{EE}$ of Case 1 . . . . . 49

5.7 Actual $z_{EE}$ of Case 1 . . . . . 49
5.8 Desired and actual $x_{EE}$ for Case 1 . . . . . 50
5.9 Desired and actual $y_{EE}$ for Case 1 . . . . . 50
5.10 Desired and actual $z_{EE}$ for Case 1 . . . . . 51
5.11 Tracking error of end-effector position for Case 1 . . . . . 52
5.12 Desired and actual end-effector trajectory in XY plane for Case 1 52
5.13 Desired and actual end-effector trajectory in the 3D plane for Case 1 . . . . . 53
5.14 Desired and actual mobile base velocity in x direction for Case 1 54
5.15 Desired and actual mobile base velocity in y direction for Case 1 55
5.16 Actual velocities of the operational space joints for Case 1 . . 56
5.17 Controlled velocities of the operational space joints for Case 1 56
5.18 Error velocities of the operational space joints for Case 1 . . . 57
5.19 Estimated parameters for Case 1 . . . . . 57
5.20 Input torque for Case 1 . . . . . 58
5.21 Input torque (5 sec) for Case 1 . . . . . 58
5.22 Desired end-effector position trajectories for Case 2 . . . . . 59
5.23 Desired and actual $x_{EE}$ for Case 2 . . . . . 60
5.24 Desired and actual $y_{EE}$ for Case 2 . . . . . 60
5.25 Desired and actual $z_{EE}$ for Case 2 . . . . . 61
5.26 Tracking error of end-Effector position for Case 2 . . . . . 61
5.27 Actual and desired trajectory tracking of the Case 2 in XY plane 62
5.28 Actual and desired trajectory tracking of the Case 2 in XY plane 63
5.29 Actual and desired trajectory tracking of the Case 2 in XZ plane 63
5.30 Desired and actual $\dot{x}$ for Case 2 . . . . . 64
5.31 Desired and actual $\dot{y}$ for Case 2 . . . . . 64
5.32 Actual velocities of the operational space joints for Case 2 . . 65
5.33 Controlled velocities of the operational space joints for Case 2 65

5.34 Error velocities of the operational space joints for Case 2 . . . 66
5.35 Estimated parameters for Case 2 . . . 66
5.36 Input torque of the joints for Case 2 . . . 67
5.37 Actual and desired x positions for Case 3 . . . 67
5.38 Actual and desired y positions for Case 3 . . . 68
5.39 Tracking errors for Case 3 . . . 68
5.40 Actual and desired velocities of mobile base in x direction for Case 3 . . . 69
5.41 Actual and desired velocities of mobile base in y direction for Case 3 . . . 69
5.42 Actual joint velocities for Case 3 . . . 70
5.43 Controlled joint velocities for Case 3 . . . 70
5.44 Error of joint velocities for Case 3 . . . 71
5.45 Estimated paramters for Case 3 . . . 72
5.46 Input torques for each joints for Case 3 . . . 72

## Abstract

A mobile manipulator is a manipulator arm mounted on the mobile platform to offer more flexibilities to reach to the target position in the space. In recent years, research activities in this area have expanded because of the mobility merges with the manipulation. A good amount of research and improvements have been carried out in the field of nonholonomic control of mobile vehicles and motion control of manipulator arms. In literatures, control methods such as state-feedback, output-feedback, dynamic coupling, model-based, adaptive tracking control, etc. have been applied to the control of mobile manipulators.

One of the objectives of this research work is to construct the systematic modeling of kinematics and dynamics of the mobile manipulator using the Lagrangian dynamics under the nonholonomic constraints. Then design a controller for n+m degree of freedom mobile manipulator with the aim of simultaneous control of the velocity of the mobile platform and the motion of the end-effector. Using the idea of kinematic backstepping control and adaptive torque control, a two-step control is presented for the nonholonomic mobile manipulator. The kinematic velocity control is designed in the first step such that all the desired trajectories are achieved. In the second step, the adaptive torque controller based on the dynamics of the mobile manipulator is designed such that the mobile platform velocity and the end-effector position converge to the reference trajectories designed in the first step. The parameters in this case are assumed to be completely unknown. The parameter update law is formed and used along with the designed controller to update the parameters. Those parameters are used as the estimates of the real parameters and a control signal is produced. This control scheme provides an efficient solution to the motion control problem. System dynamics is modeled in the MATLAB/Simulink and an appropriate controller is modelled for the specified task. Simulink results validate that the designed control method guarantees that the mobile manipulator states converge to the desired trajectories.

## List of Nomenclature

| Symbol | Description |
|---|---|
| $[x^{'}, y^{'}, z^{'}]$ | unit vectors |
| Matrix $R$ | rotational matrix |
| $l_i$ | link length of the $i^{th}$ link |
| $d_i$ | link off-set of the $i^{th}$ link |
| $\theta_i$ | joint angle of the $i^{th}$ link |
| $\alpha_i$ | link twist of the $i^{th}$ link |
| $C_{vi}$ | $\cos(v+i)$ |
| $S_{vi}S_{\alpha i}$ | $\sin(v+i)\sin(\alpha+i)$ |
| $T$ | homogeneous transformation matrix |
| $J$ | Jacobian matrix |
| $J_p$ | Jacobian matrix related to the mobile platform |
| $J_o$ | Jacobian matrix related to the manipulator arm |
| $f_c(q,t)$ | constraint function |
| $Q$ | numbers of holonomic constraints in the system |
| $P$ | numbers of generalized coordinates in the system |
| $v_c$ | contact velocity of the wheel on surface |
| $x_{EE}$ | end-effector position of the mobile manipulator in x direction |
| $y_{EE}$ | end-effector position of the mobile manipulator in y direction |
| $z_{EE}$ | end-effector position of the mobile manipulator in z direction |
| $x$ | position of mobile platform in x direction |
| $y$ | position of mobile platform in y direction |
| $l_1$ | length of the first link of the manipulator |
| $l_2$ | length of the second link of the manipulator |
| $\theta_b$ | heading angle of the mobile platform and mobile manipulator in the task space |

| Symbol | Description |
|---|---|
| $\theta_1$ | joint angle of the first link of the mobile manipulator |
| $\theta_2$ | joint angle of the second link of the mobile manipulator |
| $v$ | linear velocity of the mobile platform |
| $\omega$ | angular velocity of the mobile platform |
| $X$ | end effector position vector |
| $\eta$ | mobile manipulator joint velocities |
| $\theta_R$ | angular position of right wheel of the mobile platform |
| $\theta_L$ | angular position of left wheel of the mobile platform |
| $K$ | kinetic energy |
| $P$ | potential energy |
| $\tau$ | torques of the joints |
| $M(q)$ | inertia matrix |
| $C(q)$ | centripetal and coriolis matrix |
| $G(q)$ | gravitational force vector |
| $B(q)$ | input transformation matrix |
| $f$ | generalized constraint force vector |
| $M_b$ | inertia matrix for the mobile platform |
| $M_a$ | inertia matrix for the manipulator arm |
| $M_{ab}$ | coupling inertia matrix for the mobile manipulator |
| $M_{ba}$ | coupling inertia matrix for the mobile manipulator |
| $C_b$ | centripetal and coriolis matrix for the mobile platform |
| $C_a$ | centripetal and coriolis matrix for the manipulator arm |
| $C_{ab}$ | coupling centripetal and coriolis matrix for the mobile manipulator |
| $C_{ba}$ | coupling centripetal and coriolis matrix for the mobile manipulator |
| $G_b$ | gravitational force vector for the mobile platform |
| $G_a$ | gravitational force vector for the manipulator arm |
| $B_b$ | input transformation matrix for the mobile platform |
| $B_a$ | input transformation matrix for the manipulator arm |
| $\tau_b$ | torques of the wheels joints for the mobile platform |
| $\tau_a$ | torques of the link joints for the manipulator arm |

| Symbol | Description |
|---|---|
| $q$ | generalized coordinates of the mobile manipulator system |
| $q_b$ | generalized coordinates of the mobile platform |
| $q_a$ | generalized coordinates of the manipulator arm |
| $\lambda_n$ | non-holonomic constraints vector |
| $\lambda_h$ | holonomic constraints vector |
| $\alpha$ | steering velocity of kinematics |
| $R$ | radius of the wheels of the mobile platform |
| $D$ | distance between the wheels |
| $Y$ | regressor matrix |
| $p$ | unknown parameters |
| $\Psi$ | vector of velocity of the platform and end-effector position of the mobile manipulator |
| $\Psi_d$ | vector of desired velocity of the platform and desired end-effector position of the mobile manipulator |
| $\dot{\Psi}_b$ | vector of mobile platform velocities |
| $\dot{\Psi}_{bd}$ | vector of desired mobile platform velocities |
| $\dot{\Psi}_a$ | vector of end-effector position of the mobile manipulator |
| $\dot{\Psi}_{ad}$ | vector of desired end-effector position of the mobile manipulator |
| $e$ | error between the desired and actual vector of velocity of the platform and end-effector position of the mobile manipulator |
| $\eta_c$ | controlled joint velocities |
| $\Delta$ | decoupling matrix |
| $k$ | damping factor |
| $K$ | control gain for the kinematic controller |
| $K_1$ | control gain for the adaptive torque controller |
| $\hat{p}$ | vector of estimated parameters of the system |
| $e_p$ | error between actual and estimated parameters |
| $\Gamma$ | symmetric positive definite gain matrix |
| $\eta_e$ | error between the actual and controlled joint velocities |

## Acknowledgements

This acknowledgement is intended to be thanks giving gesture to all those people who have been involved directly or indirectly with my research project.

First and foremost, I express my special thanks with gratitude and great respect to my valuable supervisor Dr. Ya-Jun Pan for her keen interest in my research study, fruitful suggestions and valuable guidance with motivation and constant encouragement to complete project. It has been an honor to be her student. I am also thankful to her for her great patience, constructive criticism and useful suggestion apart from invaluable guidance. I am also thankful to my friend Dhaval Chaurasia for his encouragement. I am also thankful to all ACM lab members.

For this thesis, I would like to thank my oral defense committee members, Dr. Marek Kujath and Dr. Serguei Iakovlev, for their time and insightful questions.

Lastly, I would like to thank my family and my friends, Hardik Patel and Jay Chauhan for all their love and encouragement. And most of all for my loving and encouraging fiancee Shruti whose faithful support during the final stages of this Masters is so appreciated. Thank you.

I will keep on trusting you for my future. Thank you, God.

# Chapter 1

# Introduction

## 1.1 Background

A mobile manipulator is a robotic manipulator arm mounted on the wheeled mobile platform [6]. Generally, robotic manipulators are held in a fixed place and are not capable of traveling from their root location. Consequently, the workspace where the end-effector of the fixed base manipulator can reach is a limited volume of the space. If the manipulator is mounted on a mobile base, the workspace of the manipulator can be expanded due to the physical movement of the mobile platform along with the manipulator arm. This capability of the mobile manipulator gives the advantages in the different fields like military, medical, education and mining to perform the task that can be difficult for the stationary manipulator arms. It can be useful in the pick and place application in the small workspace like a room or at medium warehouses to accomplish the material handling task and other similar tasks. Each robot is different and may have problems unique to it. The robot can only do a specific task at a time. Mobile Manipulator term here refers to the terrestrial mobile manipulator and not the water or space locomotion. Differential-drive wheeled mobile manipulators and car-like mobile manipulators are simple and reliable wheel-based propulsion system which is commonly used in smaller mobile robots. When the mobile platform is a wheeled base subjected to nonholonomic constraints, the robotic system is called nonholonomic mobile manipulator. In most literature, mobile manipulator control problems are treated under the assumptions that the precise knowledge of the dynamics of the mobile manipulator is available or the kinematic constraints are ignored or the interaction between the vehicle and manipulator is ignored.

In the past three decades, robotics control has been become more interesting field of research. Robotic control field can be considered as a combination of multiple engineering disciplines such as mechanical, electrical and electronics. It requires a wide knowledge of the mechanics and electrical part of a robot with the strong capability to

model its performance. Further, it also requires strong mathematical and computer programming skills to achieve the control task. Therefore this inter-disciplinary field requires a broad knowledge. The main features of mobile manipulators include that mobile manipulators use mathematical theories. They also use coordination and kinematics in their operations. The next section discusses how these mobile manipulators work and the way in which they are controlled to answer the questions like how the mobile manipulators are designed and how is kinematics used in the control and how mobile manipulators are controlled.

## 1.2 Literature Review

Mobile manipulators are advantageous because they usually adapt to the different environments that they are provided with and that they can coordinate together to perform hard and complex jobs [7]. Here, the robotic arms and the wheels are controlled by a computer enabling them to move accordingly. Hence there is a need for assuring stability of the mobile manipulator. In the case of problems with the design of the mobile robot manipulator, the kinematics problem is usually altered to enable the robot to become more efficient [8]. Kinematics is very important in the movement process of a mobile manipulator. Because of the rolling without slipping conditions, nonholonomic mobile base is subjected to nonintegrable differential constraints [9] [10]. Planning and controlling methods have been proposed for the nonholonomic mobile platforms in following literature [11] [12]. On the contrary, fixed manipulators are kinematically unconstrained. However, the loss of mobility occurs at singular configuration for the end-effector velocity of the fixed manipulator. A prevalent approach to manage singularities is to adopt a kinematically redundant manipulator with a number of DOFs larger than the number of variables (i.e. joints variable) needed to achieve a given task [10]. For example, for $n$ DOFs manipulator, if the numbers of variables are $s$, then the $n - s$ extra DOFs can be used for optimization of performance criteria or may be helpful in accomplishing additional task requirements while performing a primary task [13]. This is also called as degree of redundancy [10]. The other approach to overcome the singularity is by adding mobility to the fixed manipulator base. In inverse kinematics, analytical and numerical approaches are used to solve problems [8]. The mathematical solution of an equation

will not always fall in line with the physical solutions. Hence the solution for each machine is always different. Each problem uses a specific formula for it to be solved.

Motion planning allows for robots to plan the trajectory of their manipulators. This gets more complex when it comes to robots with greater degrees of freedom [14]. One of the first attempts to overcome this problem is aimed at maneuvering through two-dimensional spaces where the geometrical properties of the workspace can be exploited to come up with deterministic approaches, such as vertical cell decomposition and shortest path roadmap [15]. The vertical cell decomposition involves collision-free path in which roadmap vertices are created within trapezoid-shaped cells. The collision-free path can be calculated using graph-based algorithms. The shortest path roadmap utilizes graph nodes placed at obstacle apexes and an edge exists only in the event a pair of apexes is mutually visible. In this approach, feasible paths may touch obstacles, already accounted for during obstacle modeling. These methods, however, become ineffective when applied to high-dimensional configuration spaces. Despite this, they can be used as a part of complex algorithms. Another approach is the creation of a probabilistic roadmap (PRM). It consists of a graph with nodes marking a point in a free space and a pair of points can be connected if a straight-line movement between the nodes is possible without collision. With this approach, one can find a motion plan by sampling the points in the free space to find a collision-free connection. Sample-based planning methods mostly achieve resolution completeness and are commonly used in industry-grade problems. They can be divided into a multi-query and a simple query. The latter generates a roadmap at the beginning to represent the connectivity of the configuration space then later search requests can be processed based from this. The former dies not build a roadmap at the beginning but generates a branched graph when searching for a solution. The graphs grow and are able to connect the initial and final configurations of one problem presented. This is known as the rapidly-exploring dense tree (RDT).

Mobile manipulators exhibit a profile consisting of a robotic manipulator and a mobile platform. It utilizes the manipulation capability of a fixed-based manipulator and mobility in utilizing a mobile platform, offering more flexible material handling. The leader-follower mobile manipulator is a coordination scheme where one or a group of mobile manipulators act as a leader by tracking a preplanned trajectory and the

other mobile manipulators act as followers by moving in a coordinated fashion with the lead mobile manipulator [16]. There may be a parameter uncertainty in the modelling process of the robot which can be overcome by applying a decentralized control law to individual robots. Hybrid position-force control can also be used in a decentralized/centralized control scheme [16]. In this scheme, the object is controlled in a certain direction of the workspace, with internal forces of the object controlled in a small range of the origin. Robust adaptive controllers of mobile manipulators operating in a coordinated manner with a single load have been investigated with unknown parameters and disturbances. Coupled dynamics have also been investigated, presenting two cooperating mobile manipulators controlling an object with relative motion. This was performed in the presence of uncertainties and external disturbances. Centralized robust adaptive controllers are usually introduced in such a configuration to guarantee motion and force trajectories of the load. A simulation study investigating the decentralized dynamic control for a robot collective with non-holonomic wheeled mobile manipulators by tracking trajectories of the load was performed. Reference signals were used for each robot, one for the mobile platform and another for the end-effector of the manipulating arm. Centralized control approaches may present intrinsic problems. The main problems are the difficulty to maintain communication between the robot and limited scalability of the configuration. Despite this, it can prove useful and have a technical benefit when applied to a control group of robots. This was proposed in a simulation with a centralized multi-robot system in an escort mission. The escorted agent was placed in the centroid of a polygon of n sides surrounded by n robots at the vertices of the polygon [16]. Proposed controls are flexible and robust when exposed to uncertainty and external disturbances. There is a popular and relatively simple control technique based on feed back control, that is feedback linearization. Moralez et. al considered a problem formulation where linearization of a general affine nonlinear control system was attained through the change of coordinates on the mobile manipulator [17]. It was generalized by Brockett to solve a linearization problem using approaches exploit capabilities for nonholonomic mobile manipulators [18]. One drawback of this approach is that the involutivity condition has to be maintained throughout operation, thereby limiting the number of classes that can utilize this model. It is also limited in its ability to ensure stable operation

when the nonlinear system exhibits zero dynamics. Despite this, it is possible to extend the feedback linearization to nonminimum phase systems. This was illustrated in the model based on fabricating a minimum phase approximation of the original model using inner-outer factorization. Generalized feedback linearization (GFL) is another non-linear control technique closely related to feedback linearization, the difference being that it addresses the restrictions of feedback linearization [19]. The additional denominator in the GFL control law prevents the controller from cancelling the system zero dynamics. Backstepping is a technique that can be used in systems with a specialized structure. It attempts to solve the nonholonomic navigation drawbacks that prevent tracking of a reference trajectory, following a path and point stabilization [20] [21]. It utilizes a Lyapunov design based on virtual controls [22]. Robust backstepping leads to a quadratic robust control Lyapunov functions (rclf) in a set of transformed coordinates [23] [24]. This type of rclf can generate control laws with local gains that are not entirely necessary, leading to excessive control efforts such as high magnitude chattering in the control signal. This property is amplified at every stage in recursive backstepping design. This is evident in comparison with simulation results of a second-order uncertain system under robust feedback control laws created using quadratic and flattened rclf. This comparison shows that the chattering is wasteful as the trajectories in the state space were nearly identical. Assumptions regarding smoothness can be made in recursive backstepping models as there is a need to calculate function derivatives during the creation of control law and Lyapunov function [25]. In nonlinear systems, it becomes difficult to satisfy such assumptions. Some models using backstepping designs lead to construction of rclfs for strict feedback systems, assuming a perfect state feedback.

A backstepping tracking controller developed using a hybrid neural network for a nonholonomic mobile manipulator with dual arms to achieve precise velocity and position tracking with uncertainties and disturbances was tested [26]. This configuration was contained the sliding mode neural network controller, the robust controller and the proportional controller. This is found to guarantee close-loop asymptomatically stability while restraining the system behavior in the sliding surface in the event of system uncertainties [26]. The simulation displays feasibility and effectiveness when utilizing such a configuration. Initially, adaptive control is proposed for

trajectory/force control of mobile manipulators with either holonomic or nonholonomic constraints with unknown parameters. Researchers have investigated whether the trajectory and force tracking controls of nonholonomic mobile manipulators with unknown inertia parameters, constraints and disturbances are effective and achieve the desired results [27]. Such controls ensured the output of the dynamic system tracked the variable signals and effectively stabilized the system depending on the desired force or motion [28]. This was further simulated on the control of a two wheel driven mobile manipulator and indicated that the proposed model was effective.

Adaptive control schemes can be developed based on two types of designs: direct and indirect designs [29]. The direct adaptive control system has control parameters that are directly drawn form an adaptive law that tracks errors. An indirect adaptive control system, on the other hand, has controller parameters that are calculated simultaneously using on-line estimates updated from a parameter estimator. Robust adaptive control is proposed for mobile manipulator system in an environment exhibiting parametric uncertainties and external disturbances [30]. This is created to drive the task space desired end-effector and platform trajectories with respect to nonholonomic constraints. Estimation of the unknown parameters and external disturbances are done with the use of an update law in the adaptive control scheme [31]. The simulation of the proposed configuration proves successful, with future focus on real implementation of the mobile manipulator system. Fang et.al proposed adaptive control strategies presented to control coordinated multiple mobile manipulators carrying a common object with an external environment having uncertainties and disturbances [32]. All control strategies are created to drive the system motion towards the desired manifold. The proposal involved the development of the dynamics of inter-connected system that included the coordinated mobile manipulator dynamics, object dynamics and the interaction between object environments [33]. The adaptive control law based on dynamic model and simulation lead to the conclusion that non-regressor based controls are effective and could the configuration could be applied in an external environment. Adaptive iterative learning control schemes have also been proposed for trajectory tracking of robot manipulators performing repetitive tasks under unknown parameters [34]. An iterative term is added to modify the behavior of the configuration in the presence of unknown parameters and disturbances. This

configuration is rigid but shows the use of adaptive control on the base of a mobile manipulator in overcoming the position tracking problem [35]. Proof of convergence was based on use of Lyapunov-like positive definite sequences which decreased under the proposed control scheme [36].

PID feedback control schemes have been implemented in combination with an input learning scheme, with proposals aiming for the learning of periodic robot motion [37]. The PID feedback controller, located in the learning controller, stabilizes the response of robot dynamics. Simultaneously, the learning controller computes the desired torques in the actuator for feedforward nonlinear dynamics compensation in its steady state. The error signals in the learning control system show that the motion trajectory converges to the desired on asymptomatically, ensuring its effectiveness in stabilizing the mobile manipulator and having significant gains in tracking and convergence rate when the learning system is implemented in such a configuration. Model based schemes have also been explored in regards to mobile manipulators. Some researchers used a model-predictive trajectory-tracking control that utilizes linearized tracking error-dynamics predicting future system behavior [38]. The controller proposed has velocity and acceleration constraints while a smith predictor was used to compensate for vision-system dead time. Another approach involved the modelling and control of a spatial mobile manipulator, using dynamic model based approach [39] has been proposed. The assumption in this model is that there is a perfect contact between the wheels and the ground and model is obtained using the Lagrange-dAlembert formulation. Kinematic redundancy in this case is resolved by decomposing the mobile manipulator into two subsystems, namely the mobile platform and the manipulator. This is coupled with a robust interaction control algorithm. Wheel slip is a drawback in this model as it can adversely affect the tracking performance but the robust control approach is used to minimize this effect to a greater degree. Simulation of this configuration shows its algorithm can be effective. Another simulation involved a kinematic model and dynamic model, using this model based approach to control and omnidirectional mobile manipulator [40]. The nonlinear control design was studied based on these models and utilized an input-linearization method. An impedance control is added to this configuration so as to allow for the unification of the control structure in the main case of robot interaction within the environment. It also solves

instability problems. Simulation presents good behavior when the system was met with obstacles, with respect to the different values of impedance.

Typically, nonholonomic mobile manipulator systems can be described by a kinematic model with a velocity inputs. These velocity inputs are the available commands when applying a closed loop controller to the robotic system. The kinematic model is suitable for planning and control problems therefore once a kinematic controller has been developed, it is easy to design a corresponding dynamic controller with torque inputs [41]. Two main strategies are mostly used for the feedback control of the mobile manipulators to perform a given task. Kinematic output tracking method using the error between the actual and desired output trajectory to control the trajectory tracking. The Jacobian is associated with the kinematic control and because the nonholonomic mobile manipulators are redundant with respect to the given tasks, the controller needs to include a null-space actions. This can be done by local optimization techniques with one of the objectives of being singularity avoidance [42]. The other method is state tracking method. In this method a reference trajectory is associated to the desired trajectory and then the error between the actual and desired trajectory is used to track the desired trajectory. The problem is then fixed by using existing stabilization methods. State tracking method gives a good control of the whole configuration of the nonholonomic mobile manipulator. Furthermore, output tracking methods are easier to design but cannot satisfy the accurate control. The aim of this thesis is to present a kinematic velocity control for the differential drive nonholonomic mobile manipulators and then design a suitable dynamic torque controller.

## 1.3 Thesis Contribution

The following can be expressed as novel contribution presented in this thesis:

1. This thesis presents a systematic mathematical modeling of kinematics and dynamics of a 5-DOF mobile manipulator system.

2. This thesis presents a novel control strategy for the nonholonomic mobile manipulator for n-DOF mobile manipulator system with mathematical derivation and simulation studies.

## 1.4 Thesis Organization

The thesis is organized in 6 chapters:

Chapter 1 explains the brief background of the mobile manipulator systems. It presents previous work in the field of modeling and control of mobile manipulators. Different control techniques for mobile manipulators are discussed featuring kinematic control, nonlinear control strategies, model based, backstepping, adaptive, robust adaptive methods.

In chapter 2, the detailed kinematics and dynamics modeling of a mobile manipulator is presented. The forward and inverse kinematics, Jacobian method, mobile manipulator system description, kinematic constraints are discussed briefly. Finally, properties of dynamic systems are mentioned.

Chapter 3 presents the details of designed control strategy for the $n$ DOF mobile manipulator. Firstly, the problem formulation and control objective are presented followed by the assumptions on the system. Then the designed controller is explained with mathematical proof. Redundancy resolution method is also introduced in the control design.

In chapter 4, Model based control is presented for the control of holonomic constrained nonholonomic mobile manipulator under the assumptions of known parameters and no external or internal disturbance. Control strategy is presented with the block diagram. The detailed simulation studies on model based controller is presented with results.

Chapter 5 analyses the simulation results on the backstepping adaptive aontrol. This chapter presents an extensive simulation work and results for the proposed controller applied on a 5 DOF nonholonomic mobile manipulator. System model and control scheme is presented in the Simulink environment. Two cases are presented with the two different trajectories to verify the effectiveness of the proposed controller. Plots are shown and compared to justify the results.

Chapter 6 presents the summary on the research and suggestions on future work.

# Chapter 2

# Modeling of Mobile Manipulators

## 2.1 Kinematics Fundamentals

### 2.1.1 Kinematics

Kinematics deals with the motion of the system without considering the forces which are responsible for that motion. Kinematics deals with the position, velocities and acceleration. The description of the motion is relative to the reference frame and hence position, velocity, acceleration and other higher derivatives are defined with respect to the reference frame.

#### Position and orientation of a rigid body

Position and orientation of the object or point in Cartesian coordinates or space are a familiar mathematical concept for the design and analysis of the robotic manipulators. Components of the manipulator like links, joints and tools as well as the objects which are in the surrounding of the manipulator are being traced regarding to the reference frame in order to analyze the behavior of the kinematics. Position and orientation of a rigid body are completely described in the space with respect to a reference frame. As shown in the Fig.2.1, let $O - xyz$ be the reference frame and $x$, $y$ and $z$ are the unit vectors of the reference frame axis. The position of a point $A$ on the rigid body with respect to the reference frame can be expressed by Eq.(2.2), where, $\overrightarrow{OA}_x$, $\overrightarrow{OA}_y$ and $\overrightarrow{OA}_z$ are the $x$, $y$ and $z$ direction components of the vector $\overrightarrow{OA}$.

$$\overrightarrow{OA} = \overrightarrow{OA}_x + \overrightarrow{OA}_y + \overrightarrow{OA}_z \tag{2.1}$$

From the above equation, the position of $\overrightarrow{OA}$ can also be written as a $(3 \times 1)$ vector

$$\overrightarrow{OA} = \begin{bmatrix} \overrightarrow{OA}_x \\ \overrightarrow{OA}_y \\ \overrightarrow{OA}_z \end{bmatrix} \tag{2.2}$$

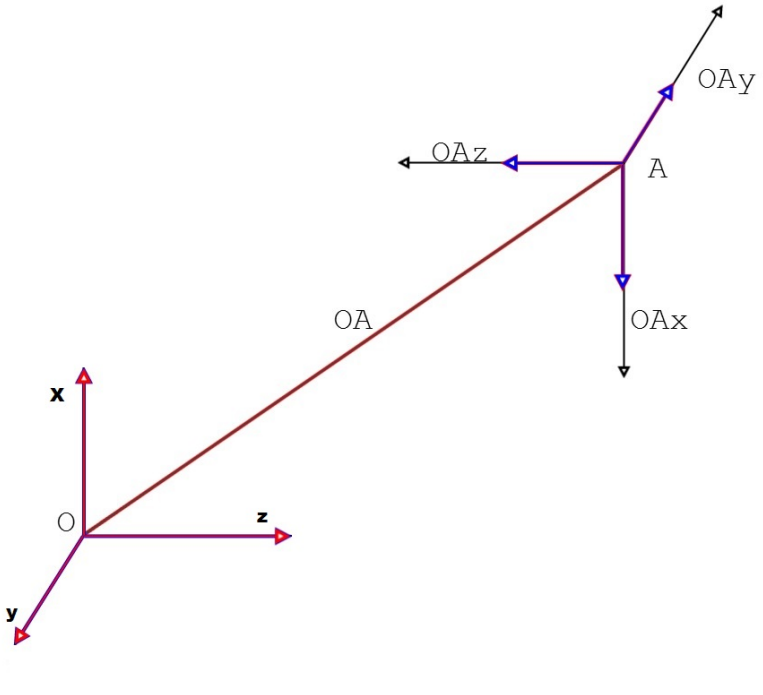


Figure 2.1: Position and orientation of a rigid body

The vector $\overrightarrow{OA}$ is bound as it is fixed in position. Bound vector deals with the forces whose point of action cannot be shifted. In contemplation of describing the rigid body orientation, it is favorable to deal with the orthonormal frame attached to the rigid body and express their unit vectors with respect to the original reference frame [43]. These vectors are expressed with respect to the reference frame $O - xyz$ by the following equations,

$$\begin{aligned} OAx = x^{'} &= x^{'}_{x}x + x^{'}_{y}y + x^{'}_{z}z \\ OAy = y^{'} &= y^{'}_{x}x + y^{'}_{y}y + y^{'}_{z}z \\ OAz = z^{'} &= z^{'}_{x}x + z^{'}_{y}y + z^{'}_{z}z \end{aligned} \tag{2.3}$$

**Rotation Matrix**

Simplifying Eq.(2.3) which describes the rigid body's orientation with respect to the reference frame in a $3 \times 3$ matrix form,

$$R = \begin{bmatrix} x^{'} & y^{'} & z^{'} \end{bmatrix} = \begin{bmatrix} x^{'}_{x} & y^{'}_{x} & z^{'}_{x} \\ x^{'}_{y} & y^{'}_{y} & z^{'}_{y} \\ x^{'}_{z} & y^{'}_{z} & z^{'}_{z} \end{bmatrix} = \begin{bmatrix} x^{'T}x & y^{'T}x & z^{'T}x \\ x^{'T}y & y^{'T}y & z^{'T}y \\ x^{'T}z & y^{'T}z & z^{'T}z \end{bmatrix} \tag{2.4}$$

Eq.(2.4) is termed as the rotation matrix and it is an orthogonal matrix satisfying

$$R^T R = I,$$

where $I$ is a $3 \times 3$ identity matrix.

### 2.1.2 Forward Kinematics and Inverse Kinematics

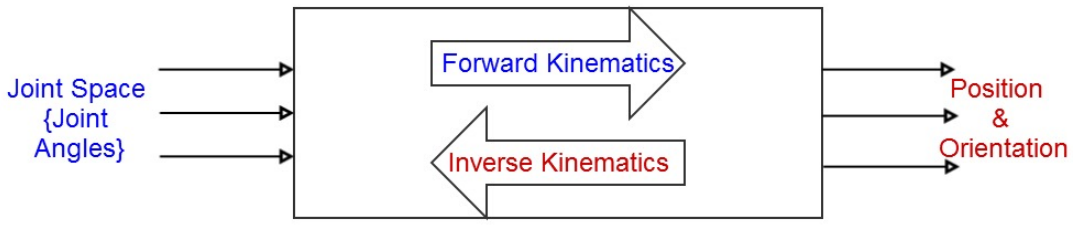


Figure 2.2: Forward and inverse kinematics [1]

The robot kinematics can be classified in two categories: forward and inverse kinematics. The forward kinematics is straightforward and the equation derivation is not much complex [44]. Inverse kinematics is much more difficult in deriving the equations and it is also computationally expensive and time consuming in real time control of robot manipulators. Nonlinearities and singularities in the model makes the inverse kinematics more difficult. The relation between the forward and inverse kinematics is shown in the Fig.2.2.

#### Forward Kinematics

As stated earlier, a manipulator is a combination of serial links jointed together with revolute or prismatic joints from the base through end-effector. Determining the position and orientation of the end effector in term of joint variables i.e. joint angle, is called forward kinematics of the serial link manipulators. To obtain forward kinematics in organized manner, one should use a suitable kinematics model of the robotic system. The PUMA robotic arm is shown in Fig.2.3 which gives the idea of direct kinematics of the arm to reach the specified position and orientation in terms of joint angles.

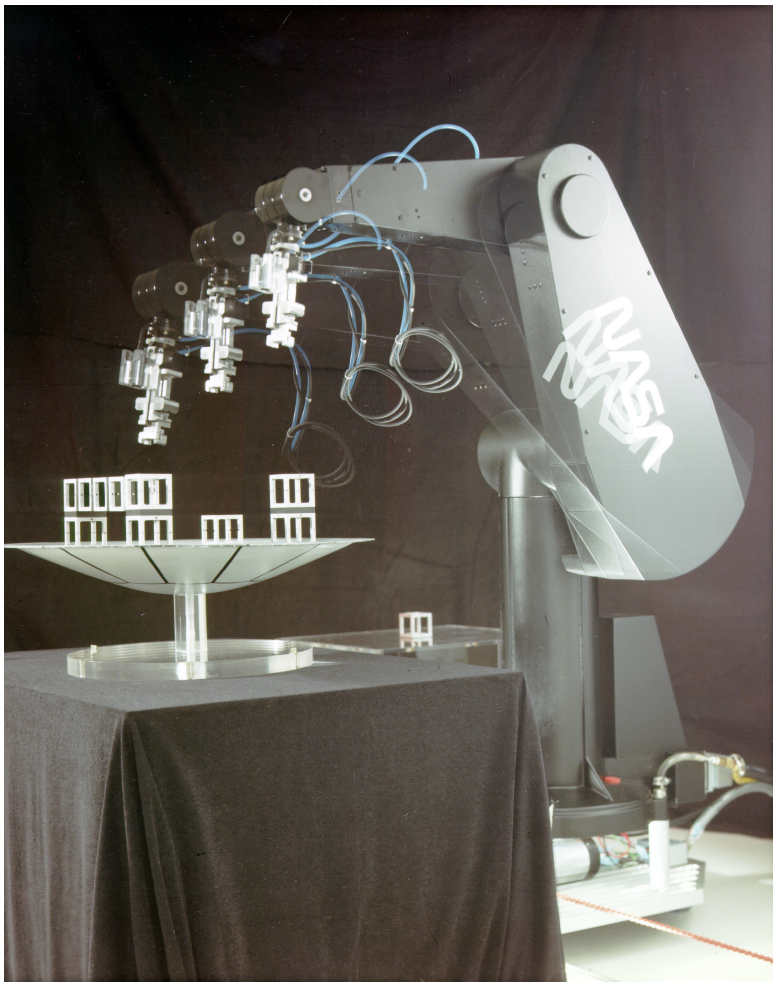

Figure 2.3: PUMA robotic arm [2]

### Denavit-Hartenberg Method

DenavitHartenberg method is the most common method used for describing the robot kinematics. It uses the four parameters link length, link twist, link offset and joint angle. A classical method to find out these D-H parameters is to systematically assign a coordinate frame to each link [45]. However there are strong constraints to place each frame because rotation of the link must be about z-axis and translation of the link must be in x-direction. Due to this it is difficult to find out the Denavit-Hartenberg parameters for the totally strange mechanism. Fig.2.4 describes the 2-link manipulator and there are four Denavit-Hartenberg parameters $a_i$, $d_i$, $\theta_i$, $\alpha_i$. They are link length, link off-set, joint angle and link-twist respectively. Using Denavit-Hartenberg approach, the resulting coordinate transformation is obtained by Eq.(2.5).

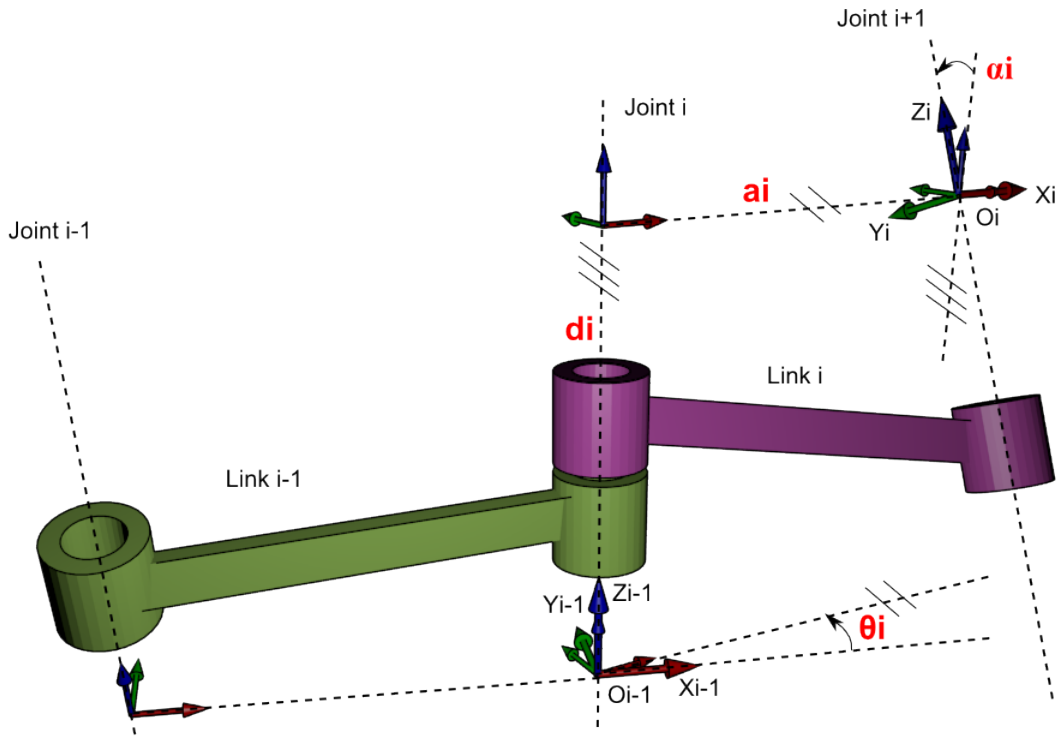


Figure 2.4: Classic D-H Parameters [3]

$$A_i^{i-1}(q_i) = \begin{bmatrix} c_{v_i} & -s_{v_i}c_{\alpha_i} & s_{v_i}s_{\alpha_i} & a_i c_{v_i} \\ s_{v_i} & c_{v_i}c_{\alpha_i} & -c_{v_i}v_{\alpha_i} & s_{v_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.5}$$

It is clear from the above equation that the transformation matrix is the function of the joint variable, that is, either $d_i$ for the prismatic joint or $v_i$ for the revolute joint. The Denavit-Hartenberg convolution grants formulating the direct kinematics function by the composition of each transformation expressed in Eq.(2.5) into one homogeneous transformation matrix.

$$T_n^0(q) = A_1^0(q_1)A_2^1(q_2)A_3^2(q_3)....A_n^{n-1}(q_n). \tag{2.6}$$

The calculation of forward kinematics function is periodic and can be obtained in an organized form by simple products of the all matrices. This method can be used for any open loop serial manipulator. Eq.(2.6) is the homogeneous forward kinematics equation.

**Inverse Kinematics**

The conversion of the position and orientation of the end effector of the manipulator from Cartesian space to the joint space is called inverse kinematics of the manipulator. Cartesian space is the space which includes position and orientation of the end effector whereas joint space contains the joint variables.

There are two methods to solve the inverse kinematics problem analytically. They are geometric solution approach and algebraic solution approach. If the joint variables are known for the particular kinematics chain, the direct kinematics from Eq.(2.6) can be calculated easily in a unique way. However, the solution to inverse kinematics problem is difficult to find out due to the following reasons:

- More than one solution may exist.
- Equations are non-linear in most the cases, thus it is not possible to find the closed-loop solution all the times.
- If the manipulator is kinematically redundant, there may be infinite solutions.

### 2.1.3 Differential Kinematics and the Jacobians

In the above section, direct and inverse kinematics of the manipulators have been discussed. It shows the relationship between joint variables and the tool position and orientation. Now in this section, differential kinematics will be discussed which gives the relationship between the joint velocity and the tool linear and angular velocity. It is possible to calculate the Jacobian matrix via the differentiation of the direct kinematics function with respect to joint variables.

Forward kinematics of an n-degree of freedom manipulator can be written as Eq.(2.7)

$$T(q) = \begin{bmatrix} R(q) & p(q) \\ O^T & 1 \end{bmatrix}, \tag{2.7}$$

where $q = [q_1, q_2, q_3, ..., q_n]^T$ is the joint variable vector. Tool position and orientation changes with change in the joint vectors. The main objective of the differential kinematics is to find the relationship between the joint variable velocity and end-effector linear and angular velocities. Consider the end-effector velocity $\dot{p}$ and angular

velocity $\omega$ as a function of $\dot{q}$ joint velocities by means of the following equations,

$$\begin{aligned} \dot{p} &= J_p(q)\dot{q} \\ \omega &= J_o(q)\dot{q}. \end{aligned} \tag{2.8}$$

In Eq.(2.8), $J_p \in \mathbb{R}^{3\times3}$ is relative to the contribution of the joint velocity $\dot{q}$ to the tool linear velocity $\dot{p}$ while in second equation, $J_o$ is the $3 \times 3$ matrix relative to the contribution of joint velocities to the tool angular velocity $\omega$. These equations can be written in a more compact form as below,

$$v = \begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q)\dot{q}. \tag{2.9}$$

Eq.(2.9) represents the differential kinematics of the manipulator. Geometric Jacobian matrix is $6 \times n$ matrix and it is a function of the joint variables. Eq.(2.10) shows the Jacobian matrix,

$$J = \begin{bmatrix} J_p \\ J_o \end{bmatrix}. \tag{2.10}$$

## 2.2 System Description

Consider the (n+m)-DOF mobile manipulator system as shown in Fig.2.5. The mobile platform consists of a filled rectangular or circular plate and several wheels system. The manipulator arm consists of $n$ numbers of links and are connected together with the rotation joints. The first link of the manipulator can rotate around $z$ axis and the other links can rotate up and down. The platform of the mobile manipulator is a 2 wheeled differential drive mobile robot. The manipulator is assumed to be mounted on the center of the platform and on the midpoint of the wheel axle.

### 2.2.1 Kinematic Constraint

Mechanical systems can be classified into linear and nonlinear systems; and nonlinear systems can be further grouped as constrained holonomic systems and constrained nonholonomic systems. Constraints play an important role in governing the motion of the mobile manipulator systems. In Fig.2.5, the platform is used as the wheeled mobile robot which falls under this category.

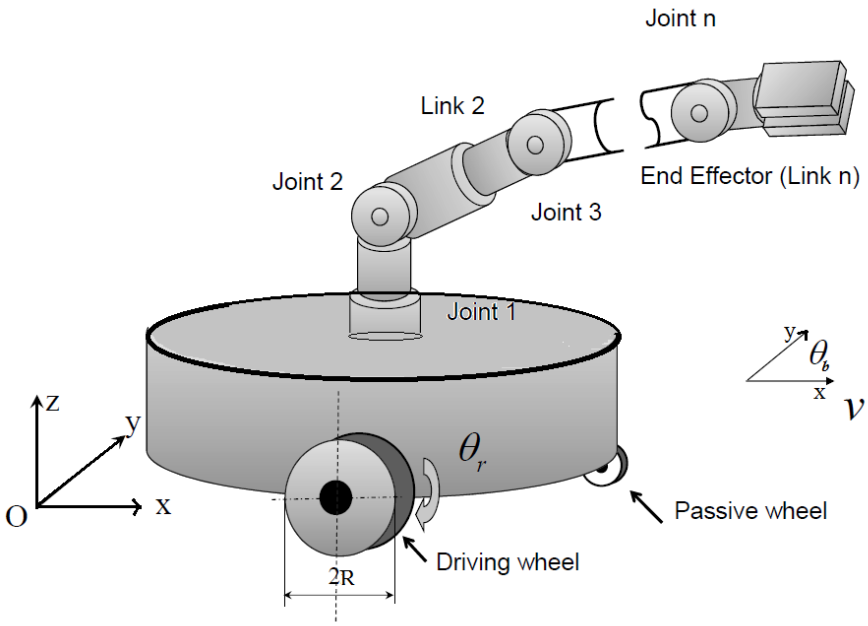


Figure 2.5: (n+m)-DOF mobile manipulator

### Holonomic Constraint

A kinematic constraint is called holonomic if it is expressed in the following form,

$$f_c(q,t) = 0, \tag{2.11}$$

where q is the vector of generalized coordinates of the system. The holonomic constraint reduces the system DOF. For example, if there is $P$ number of generalized coordinates and $Q$ number of holonomic constraints in the system, there are $PP = Q - P$ number of independent coordinates. Thus, $PP$ number of coordinates are required to describe the system and the same number of inputs are required to run the system.

### Nonholonomic Constraint

A kinematic constraint is called nonholonomic if it cannot be expressed in the form of Eq.(2.11). Nonholonomic constraints cannot be reduced to that form. In the mobile manipulator system, the wheeled mobile platform is subjected to the nonholonomic constraints. The concept and characteristics of nonholonomic constraint is presented

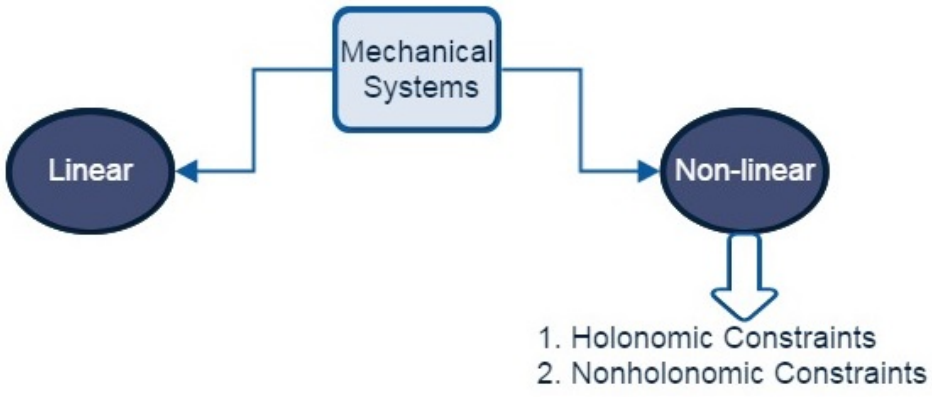


Figure 2.6: Kinematic systems

in details in [46]. In most of the general mechanical systems, nonholonomic constraints can be described by

$$f_c(q, \dot{q}, t) = 0. \tag{2.12}$$

**Assumptions**

There are several assumptions made for the wheeled mobile platform.

Wheel nonholonomic constraint

- Wheel movement is limited to horizontal plane only. It means the mobile base can only move in the $X - Y$ plane. Also, it cannot move sideways (in the direction of the axle of the wheels).

- Only point contact of the wheel to the horizontal plane. Also, it is subjected to pure rolling condition. i.e. $v_c = 0$ at the contact point.

- The friction between axle and wheel, the friction between wheel and contact surface are completely ignored.

- The steering axis of mobile platform is orthogonal to the horizontal plane.

- It is assumed that the platform of the mobile manipulator is driven by two motors independently.

### 2.2.2 Kinematic Modeling of Mobile Manipulator

Assuming a 2-link manipulator mounted on the differential drive mobile platform. Suppose $\theta_1$, $\theta_2$ and $l_1$, $l_2$ are joint angles and link lengths of the two links of the manipulator arm respectively.

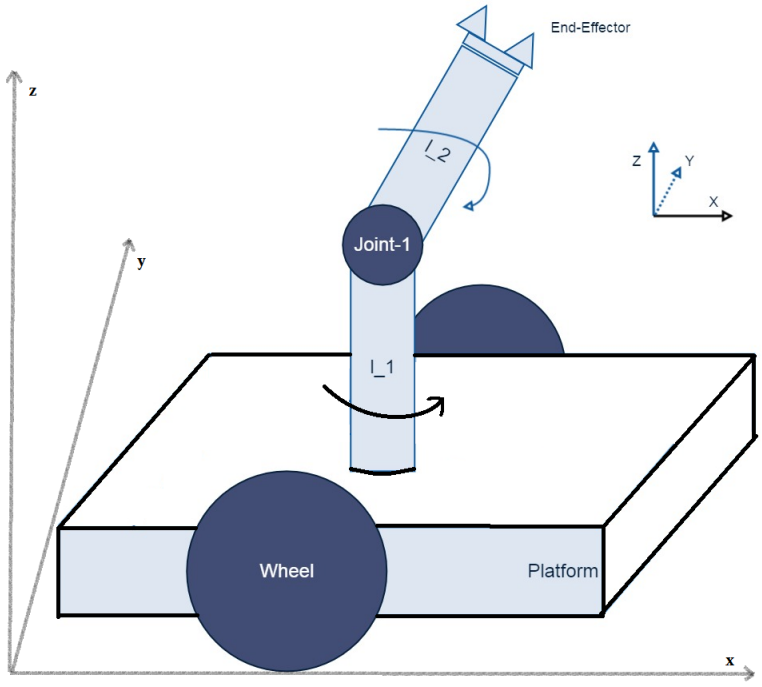


Figure 2.7: 5-DOF mobile manipulator system

The coordinates of the end-effector can be obtained from Fig.2.7.

$$\begin{cases} x_{EE} = x - l_2 \sin\theta_2 \cos(\theta_b + \theta_1) \\ y_{EE} = y - l_2 \sin\theta_2 \sin(\theta_b + \theta_1) \\ z_{EE} = l_1 - l_2 \cos\theta_2, \end{cases} \tag{2.13}$$

where $x$, $y$ is the position where the manipulator arm is mounted (also the postiion of mobile base in the coordinate system) and $\theta_b$ is the orientation of the mobile platform. Now taking the first order derivative of the end-effector coordinates,

$$\begin{cases} \dot{x}_{EE} = \dot{x} + l_2 \sin\theta_2 \sin(\theta_b + \theta_1)\dot{\theta}_b + l_2 \sin\theta_2 \sin(\theta_b + \theta_1)\dot{\theta}_1 - l_2 \cos\theta_2 \cos(\theta_b + \theta_1)\dot{\theta}_2 \\ \dot{y}_{EE} = \dot{y} - l_2 \sin\theta_2 \cos(\theta_b + \theta_1)\dot{\theta}_b - l_2 \sin\theta_2 \cos(\theta_b + \theta_1)\dot{\theta}_1 - l_2 \cos\theta_2 \sin(\theta_b + \theta_1)\dot{\theta}_2 \\ \dot{z}_{EE} = l_2 \sin\theta_2 \dot{\theta}_2, \end{cases} \tag{2.14}$$

From Eq.(2.13) and Eq.(2.14), the derivatives of the end-effector coordinates can be written in the following form

$$\begin{cases} \dot{x}_{EE} = v\cos\theta_b + l_2\sin\theta_2\sin(\theta_b+\theta_1)\omega + l_2\sin\theta_2\sin(\theta_b+\theta_1)\dot{\theta}_1 - l_2\cos\theta_2\cos(\theta_b+\theta_1)\dot{\theta}_2 \\ \dot{y}_{EE} = v\sin\theta_b - l_2\sin\theta_2\cos(\theta_b+\theta_1)\omega - l_2\sin\theta_2\cos(\theta_b+\theta_1)\dot{\theta}_1 - l_2\cos\theta_2\sin(\theta_b+\theta_1)\dot{\theta}_2 \\ \dot{z}_{EE} = l_2\sin\theta_2\dot{\theta}_2. \end{cases} \tag{2.15}$$

Considering the end-effector velocity $\dot{X}$ and mobile base linear and angular velocities and joint velocities of manipulator arm as $\eta$

$$\begin{aligned} \dot{\mathbf{X}} &= \begin{bmatrix} \dot{x} & \dot{y} & \dot{x}_{EE} & \dot{y}_{EE} & \dot{z}_{EE} \end{bmatrix}^T \\ \boldsymbol{\eta} &= \begin{bmatrix} v & \omega & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T. \end{aligned} \tag{2.16}$$

$\eta$ can be also represented in the term of right and left wheel velocities by Eq. (2.17)

$$\boldsymbol{\eta} = \begin{bmatrix} \dot{\theta}_R & \dot{\theta}_L & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^T \tag{2.17}$$

Eq.(2.15) can be expressed in the form of Eq.(2.9) in Eq.(2.18)

$$\dot{\mathbf{X}} = J\boldsymbol{\eta}, \tag{2.18}$$

where $J$ is the Jacobian matrix of the nonholonomic mobile manipulator system and the details of $J$ is given in Appendix A.1. Also, from Eq.(2.15), it is clear that the end-effector velocity is not depending on the length of the first link which rotaes around the $z$ axis.

### 2.2.3 Dynamics of Mobile Manipulators

Dynamics of the manipulator plays a very important role in the design of control algorithm, analysis of manipulator structure, and simulation of the motion. Simulation of the manipulator motion grants testing control strategies. Analysis of the dynamic model of the manipulator is very helpful in the mechanical design of the manipulator. Computation of the forces and torques required for the execution of typical motion provides useful information for the designing joints, transmission and actuators [43]. Two widely used methods for the derivation of the equations of the motion of a manipulator in the joint space are introduced. The first method is based on the closed-form

Euler-Lagrange formulation and it is theoretically easy and systematic. The second method is based on the Newton-Euler formulation and allows retrieving the model in a repetitive form. It is computationally very efficient. We used the Euler-Lagrange formulation to derive the dynamics of the system. This method deals with both the wheeled mobile base and the manipulator arm as a whole system and implement the analysis using the Lagrangian function. The Lagrangian $\mathcal{L}$ is defined as the difference between the total kinetic energy and total potential energy.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = K(\mathbf{q}, \dot{\mathbf{q}}) - P(\mathbf{q}), \tag{2.19}$$

The n-link manipulator dynamics can be written in the form given in Eq.(2.20) using Euler-Lagrange dynamic method [43],

$$\frac{du}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}_i} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}_i} = \boldsymbol{\tau}_i, \tag{2.20}$$

where, $\boldsymbol{\tau}_i$ is the corresponding torque of the $i^{th}$ joint and $i = 1, 2..., n$. Using the Euler-Lagrange method, the following dynamic equation of the mobile manipulator derived for an n-DOF manipulator mounted on the nonholonomic differential drive mobile platform, the detailed calculation of the equation is presented in Appendix A.2.

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) + \mathbf{f} = B(\mathbf{q})\boldsymbol{\tau}, \tag{2.21}$$

where $M(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix which is symmetric bounded positive definite matrix; $C(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ is the centripetal and Coriolis matrix; $G(\mathbf{q}) \in \mathbb{R}^n$ is the gravitational forces vector; $\mathbf{f} = \begin{bmatrix} \mathbf{f}_n^T & \mathbf{f}_h^T \end{bmatrix}^T = \begin{bmatrix} (A^T(\mathbf{q}_b)\boldsymbol{\lambda})^T & 0 \end{bmatrix}^T \in \mathbb{R}^n$ is the generalized constraint forces, where $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_n & \boldsymbol{\lambda}_h \end{bmatrix}^T$ is the Lagrangian multiplier with $\boldsymbol{\lambda}_n$ consider the nonholonomic constraint and $\boldsymbol{\lambda}_h$ consider the holonomic constraints. $B(\mathbf{q}) \in \mathbb{R}^{n\times m}$ is a full rank input transformation matrix and also assumed to be known; $\boldsymbol{\tau} \in \mathbb{R}^m$ is the control input to the system. $\mathbf{q} = \begin{bmatrix} \mathbf{q}_b^T & \mathbf{q}_a^T \end{bmatrix}^T \in \mathbb{R}^n$ is the vector of generalized coordinates. $\mathbf{q}_b$ denotes the generalized coordinates of the wheeled mobile base and $\mathbf{q}_a$ denotes the generalized coordinates of the manipulator

arm. The terms can be further represented as Eq.(2.22)

$$M(\mathbf{q}) = \begin{bmatrix} M_b & M_{ba} \\ M_{ab} & M_a \end{bmatrix}, \quad C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} C_b & C_{ba} \\ C_{ab} & C_a \end{bmatrix}, \quad G(\mathbf{q}) = \begin{bmatrix} G_b \\ G_a \end{bmatrix},$$
$$B(\mathbf{q}) = \begin{bmatrix} B_b & 0 \\ 0 & B_a \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_b \\ \boldsymbol{\tau}_a \end{bmatrix} \tag{2.22}$$

where $M_b$, $M_a$ describe the inertia matrices for the mobile base and manipulator arm respectively. $M_{ba}$ and $M_{ab}$ describe the coupling inertia matrices of the mobile base and manipulator arm. $C_b$, $C_a$ are the centripetal and Coriolis torques for the mobile base and manipulator arm, respectively. $C_{ba}$ and $C_{ab}$ are the coupling centripetal and Coriolis torques of the mobile base and manipulator arm. $G_b$ and $G_a$ are the gravitational force of the mobile base and manipulator arm, respectively. $B_b$ and $B_a$ denotes the input transformation matrices of mobile platform and manipulator arm, respectively. $\tau_b$, $\tau_a$ are the control input of the mobile base and robotic arm, respectively.

**Constraint Equation of Nonholonomic Mobile Plarform**

Let $\mathbf{q}_b \in \mathbb{R}^m$ and $\mathbf{q}_a \in \mathbb{R}^n$ describe the coordinates of the mobile base and the coordinates of the manipulator arm respectively. Where $m$ and $n$ is the DOF of the mobile base and manipulator arm respectively. The coordinates of the mobile base can be described by Eq.(2.23).

$$\mathbf{q}_b = \begin{bmatrix} x \\ y \\ \theta_b \end{bmatrix}, \tag{2.23}$$

where $x, y$ are the coordinated of the center of the mobile base and $\theta_b$ is the orientation or the heading angle of the mobile base. According to assumptions made for the wheels of the mobile base, the mobile base only moves in the direction perpendicular to the axis of driving wheels which means pure rolling and nonslipping movement of the mobile base. Wheels cannot slip or move in the lateral direction of the axis of the wheels. Therefore, the nonholonomic kinematic constraint for the mid point of the wheel axle where the manipulator arm is mounted can be expressed as Eq.(2.24)

$$\dot{x} \sin \theta_b - \dot{y} \cos \theta_b = 0. \tag{2.24}$$

The constraint Eq.(2.24) can also be written in the following form,

$$A(\mathbf{q}_b)\dot{\mathbf{q}}_b = 0. \tag{2.25}$$

where,

$$A(q_b) = [\sin\theta_b \quad -\cos\theta_b \quad 0] \tag{2.26}$$

Suppose there are $l$ numbers of non-integrable and independent velocity constraints and it is assumed to be have full rank $l$. The mobile platform here is assumed to be completely nonholonomic and we can write $A(\mathbf{q}_b)$ matrix of Eq.(2.25) as

$$A(\mathbf{q}_b) = \begin{bmatrix} A_1^T(\mathbf{q}_b) & A_2^T(\mathbf{q}_b) & A_3^T(\mathbf{q}_b) & ... & A_l^T(\mathbf{q}_b) \end{bmatrix}^T. \tag{2.27}$$

The nonholonomic generalized constraint forces can be given by Eq.(2.28)

$$\mathbf{f}_n = (A^T(\mathbf{q}_b)\boldsymbol{\lambda}_n)^T. \tag{2.28}$$

$H(\mathbf{q}_b) \in \mathbb{R}^{n\times m}$ is a matrix with rank being $m$ formed by a set of smooth and linearly independent vectors spanning the null space of matrix $A(\mathbf{q}_b)$, i.e.

$$H^T(\mathbf{q}_b)A^T(\mathbf{q}_b) = 0, \tag{2.29}$$

where $H(\mathbf{q}_b) = [H_1(\mathbf{q}_b), H_2(\mathbf{q}_b)..., H_{n_b-l}(\mathbf{q}_b)]$. Note that here $H^T H$ is a full rank.

According to Eqs.(2.25) and (2.29), the first order velocity kinmeatic model of a nonholonomic mobile platform which is also called the *steering system* can be written in the following form,

$$\dot{\mathbf{q}}_b = H(\mathbf{q}_b)\alpha, \tag{2.30}$$

where $\alpha$ is an auxillary function $\alpha \in \mathbb{R}^2$ and called *steering velocity* of the kinmeatic system. $\alpha_1$ and $\alpha_2$ are linear and angular velocity of the wheeled mobile platform respectively and can be written as Eq.(2.31) or $\alpha_1$ and $\alpha_2$ are the left wheel velocity and right wheel velocity of the mobile base respectively and can be written as Eq.(2.32).

$$\boldsymbol{\alpha} = \begin{bmatrix} v & \omega \end{bmatrix}^T \tag{2.31}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \theta_R & \theta_L \end{bmatrix}^T \tag{2.32}$$

Rewrite Eq.(2.30) in the specific kinematic form, in terms of linear and angular velocities of the wheeled mobile platform and in terms of right and left wheel velocities.

$$\dot{\mathbf{q}}_b = \begin{bmatrix} \cos\theta_b & 0 \\ \sin\theta_b & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2.33}$$

$$\dot{\mathbf{q}}_b = \begin{bmatrix} \frac{R}{2}\cos\theta_b & \frac{R}{2}\cos\theta_b \\ \frac{R}{2}\sin\theta_b & \frac{R}{2}\sin\theta_b \\ \frac{R}{2D} & -\frac{R}{2D} \end{bmatrix} \begin{bmatrix} \theta_r \\ \theta_l \end{bmatrix}, \tag{2.34}$$

where $R$ is the radius of the wheels and $D$ is the distance of the two wheels of the mobile platform.

Let $\eta = \begin{bmatrix} \boldsymbol{\alpha}^T & \dot{\mathbf{q}}_a^T \end{bmatrix}^T$. Due to the nonholonomic constraint defined in Eq. (2.25) and Eq. (2.30), there is an existence of a vector $\dot{\boldsymbol{\eta}}$, such that,

$$\dot{\mathbf{q}} = H(\mathbf{q})\boldsymbol{\eta}, \tag{2.35}$$

and its derivative,

$$\ddot{\mathbf{q}} = H(\mathbf{q})\dot{\boldsymbol{\eta}} + \dot{H}(\mathbf{q})\boldsymbol{\eta}. \tag{2.36}$$

Considering the nonholonomic constraints and its derivatives given in Eq.(2.35) and Eq.(2.36), the dynamics of the mobile maninpulator can be expressed by Eq.(2.37) by putting Eq.(2.35) and (2.36) into Eq.(2.21).

$$\bar{M}(\mathbf{q})\dot{\boldsymbol{\eta}} + \bar{C}(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\eta} + \bar{G}(\mathbf{q}) = \bar{\boldsymbol{\tau}}. \tag{2.37}$$

Eq.(2.37) is the reduced dynamic equation of the robotic system.

$$\begin{cases} \bar{M}(\mathbf{q}) = H^T(\mathbf{q})M(\mathbf{q})H(\mathbf{q}) \\ \bar{C}(\mathbf{q}, \dot{\mathbf{q}}) = H^T(\mathbf{q})[M(\mathbf{q})\dot{H}(\mathbf{q}) + C(\mathbf{q}, \dot{\mathbf{q}})H(\mathbf{q})] \\ \bar{G}(\mathbf{q}) = H^T(\mathbf{q})G(\mathbf{q}) \\ \bar{\boldsymbol{\tau}} = H^T(\mathbf{q})B(\mathbf{q})\boldsymbol{\tau} \end{cases} \tag{2.38}$$

A more specific dynamic model can be described by

$$\begin{bmatrix} H^T M_b H & H^T M_{ba} \\ M_{ab} H & M_a \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\alpha}} \\ \ddot{\mathbf{q}}_a \end{bmatrix} + \begin{bmatrix} H^T M_b \dot{H} + H^T V_b H & H^T C_{ba} \\ M_{ab} \dot{H} + C_{ab} H & C_a \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\alpha}} \\ \dot{\mathbf{q}}_a \end{bmatrix} + \begin{bmatrix} H^T G_b \\ G_a \end{bmatrix} = \begin{bmatrix} H^T B_b \boldsymbol{\tau}_b \\ B_a \boldsymbol{\tau}_a \end{bmatrix} \tag{2.39}$$

**Dynamic Properties**

Eq.(2.38) is the coupled dynamics of the mobile manipulator which has several properties discussed below.

**Property 1**: The inertia matrix $\bar{M}(q)$ is symmetric and positive definite matrix which follows the following inequality:

$$\lambda_{min} \bar{M}(\mathbf{q}) I \leq \bar{M}(\mathbf{q}) \leq \lambda_{max} \bar{M}(\mathbf{q}) I, \tag{2.40}$$

where $\lambda_{min} \bar{M}(\mathbf{q})$ and $\lambda_{max} \bar{M}(\mathbf{q})$ are the minimum and maximum eigenvalues of the $\bar{M}(\mathbf{q})$.

**Property 2**: There exists a skew-symmetric relationship between the inertia matrix $\bar{M}(\mathbf{q})$ and the centripetal and Coriolis matrix $\bar{C}(\mathbf{q}, \dot{\mathbf{q}})$ as follow:

$$X^T [\dot{\bar{M}}(\mathbf{q}) - 2\bar{C}(\mathbf{q}, \dot{\mathbf{q}})] X = 0. \tag{2.41}$$

**Property 3**: The left hand side of Eq. (2.37) is linearly parameterized in terms of system parameters as shown below:

$$\bar{M}(\mathbf{q}) \dot{\boldsymbol{\eta}} + \bar{C}(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\eta} + \bar{G}(\mathbf{q}) = Y(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\eta}, \dot{\boldsymbol{\eta}}) \mathbf{p}, \tag{2.42}$$

where $\mathbf{p}$ is the vector of uncertain or unknown parameters and $Y$ is a *regressor matrix* which contains unknown parameters.

# Chapter 3

# Proposed Control Design

## 3.1 Problem Formulation

### 3.1.1 Challenging Problems

1. Mobile manipulators are *kinematically redundant*. Due to kinematics redundancy, the mobile manipulators are subjected to have infinite solution for their inverse kinematics and hence subjected to *singularity* [47]. The methods used to calculate this inverse kinematics of redundant mobile manipulators are computationally complicated.

2. A wheeled mobile base considered here is a differential drive mobile platform which is subjected to nonholonomic constraints.

3. Due to different dynamics of a manipulator arm and a mobile base, there will be a dynamic interaction between both dynamics [19] which will result in complicated dynamic equations, which require more complicated and sophisticated control.

4. It is very complicated to combine the motion control problem of the wheeled mobile base and the posiotion control of the manipulator arm. Trajectory tracking task becomes more complicated and difficult to achieve due to simultaneous movement of the manipulator and wheeled mobile base.

We address the modelling and control problem of the nonholonomic mobile manipulators for the n-DOF manipulator mounted on the differential drive wheeled mobile base. The present work firstly aims at the accurate modelling of the kinematics and dynamics of the mobile manipulator using Lagrangian-Euler Dynamic method; secondly, design of a two-stage controller with the objective to track the desired position and velocity of the end-effector. In the first step of the controller design, a kinematic

control law is considered using the backstepping control method to control the velocity of the mobile manipulator. In the second step, the adaptive computed torque controller is designed which estimates the unknown parameters of the mobile manipulator system and controls the torque in order to achieve the asymptotic trajectory tracking of the end-effector.

### 3.1.2 Control Objective

The trajectory tracking problem for nonholonomic mobile manipulator can be described as follows. Let there be a mobile manipulator consist of an n-links connected together with revolute joints mounted on the nonholonomic wheeled mobile vehicle. The system can be described by the Eq.(2.37) which is subjected to nonholonomic constraints as Eq.(2.35). Those equations are rewritten here.

$$\begin{aligned} \dot{\mathbf{q}} &= H(\mathbf{q})\boldsymbol{\eta} \\ \dot{\boldsymbol{\eta}} &= \bar{M}^{-1}(\mathbf{q})[\bar{\boldsymbol{\tau}} - \bar{C}(\mathbf{q},\dot{\mathbf{q}})\boldsymbol{\eta} - \bar{G}(\mathbf{q})]. \end{aligned} \tag{3.1}$$

$\boldsymbol{\eta}$ is assumed as

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\alpha}^T & \dot{\mathbf{q}}_a^T \end{bmatrix}^T, \tag{3.2}$$

where $\alpha$ is the vector of steering velocity of the mobile platform and $\dot{q}_a$ is the vector of joint velocities of the manipulator arm mounted on the mobile base.

The controller is designed into two steps. First, motion/velocity tracking control of the end-effector of the mobile manipulator assuming that the desired trajectory is generated by kinmematic equations and then using that control as a virtual input to Eq. (2.37) and by calculating the torque control $\bar{\tau}$ that accomplish the asymptotic trajectory tracking of the mobile manipulator systems given in Eq. (3.1) for any desired trajectory $\boldsymbol{\Psi}_d = \begin{bmatrix} \dot{x}_d & \dot{y}_d & x_{EEd} & y_{EEd} & z_{EEd} \end{bmatrix}^T = \begin{bmatrix} \dot{\boldsymbol{\Psi}}_{bd}(t) & \boldsymbol{\Psi}_{ad}(t) \end{bmatrix}^T$, where $\dot{x}$ and $\dot{y}$ are the derivative of the $x$ and $y$ components of the mobile platform, and $x_{EE}$, $y_{EE}$, $z_{EE}$ are the end-effector position coordinates in respective directions. $x_{EEd}$, $y_{EEd}$, $z_{EEd}$ are the desired end-effector positions.

### 3.1.3 Assumptions

The following assumptions are made in the work presented.

1. The nonholonomic constraints and the kinematics of the platform are considered for the velocity control of the mobile platform.

2. The system is subjected to nonholonomic constraints.

3. No external forces act on the robotic system.

4. It is assumed that $\Psi_d(t)$ is computable with $\dot{\Psi}_{bd}(t)$. In other words mobile manipulator is able to track the $\Psi_d$ and $\dot{\Psi}_{bd}$ simultaneously and $\Psi_d$ and $\dot{\Psi}_{bd}$ are inside the range of the workspace of the mobile manipulator.

5. The desired trajectory of the end-effector $\Psi_d(t)$ of the mobile manipulator is assumed to be bounded and uniformly continuous and also assumed that it has derivatives up to the second order and they are also uniformly bounded and continuous. (However, assumed constant for the $Z_{EE}$ here.)

6. The desired velocity trajectory $\dot{\Psi}_{bd}(t)$ of the mobile manipulator is assumed to be bounded and uniformly continuous and also assumed that it has derivatives up to the second order and their derivatives are also assumed to be uniformly bounded and continuous.

7. It is assumed that the mass of the platform, mass of the links, length of the links and the inertia parameters are unknown for the mobile manipulator system. More significantly, the parameters of the mobile manipulator system are unknown.

**Barbalat's Lemma** [48]

If $f : R^+ \to R$ is uniformly continuous and positive function for all $t \geq 0$, and if the limit of integral

$$\lim_{t\to\infty} \int_0^\infty f(\tau)d\tau \tag{3.3}$$

exist and finite, then

$$\lim_{t\to\infty} f(t) = 0 \tag{3.4}$$

**Definition**

The $L_s$ norm of a function $f : R^+ \to R$ is defined as:

$$||f(t)||_s = \Big( \int_0^\infty |f(\tau)|^s d\tau \Big)^{\frac{1}{s}}. \tag{3.5}$$

for all $s = [1, \infty)$, while

$$||f(t)||_\infty = max|f(t)| \qquad t \geq 0. \tag{3.6}$$

denotes the $L_\infty$ norm of function $f$. $f \in L_s$ if $||f(t)||_s$ exist and finite. For example, $f \in L_2$ if $\int_0^\infty |f(\tau)|^2 d\tau < \infty$.

With the above given assumptions and system we can formulate the control problem as below,

### 3.1.4 Velocity Tracking Problem

Develop a control law $\boldsymbol{\eta}_c(t)$ such that all the state variables of the kinematic Eq. (2.35) for any $(\boldsymbol{\Psi}(0), \dot{\boldsymbol{\Psi}}(0)) \in \boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ and $\dot{\boldsymbol{\Psi}}$ convereg to a manifold specified as $\boldsymbol{\Phi}_d$

$$\begin{aligned} \boldsymbol{\Phi}_d &= \{(\boldsymbol{\Psi}, \dot{\boldsymbol{\Psi}}) \mid \boldsymbol{\Psi} = \boldsymbol{\Psi}_d, \dot{\boldsymbol{\Psi}} = \dot{\boldsymbol{\Psi}}_d\} \\ \boldsymbol{\Psi} &= \begin{bmatrix} \dot{x} & \dot{y} & x_{EE} & y_{EE} & z_{EE} \end{bmatrix}^T \end{aligned} \tag{3.7}$$

To simplify the control objective, assume the tracking error of the desired trajectory is $\mathbf{e}(t)$ and it can be written in the form of Eq. (3.8) where $\boldsymbol{\Psi}(t)$ denoting the configuration of mobile platform velocity and the position of the end-effector and $\boldsymbol{\Psi}_d(t)$ denotes the desired configuration of mobile platform velocity and the position of the end-effector.

$$\mathbf{e}(t) = \boldsymbol{\Psi}_d(t) - \boldsymbol{\Psi}(t) \tag{3.8}$$

The control objecive can be formulated as

$$\lim_{t\to\infty} \mathbf{e}(t) = 0 \quad \text{and} \quad \lim_{t\to\infty} \dot{\mathbf{e}}(t) = 0 \tag{3.9}$$

The basic block diagram of the designed controller is given in the Fig.(3.1).

### 3.1.5 Torque Control Problem

Using a property of the dynamic system given in Eq.(2.42) design a control law such that all the state variables of the mobile manipulator dynamics converge to a desired trajectory $\boldsymbol{\Psi}_d(t)$. In other words, design a controller $\bar{\boldsymbol{\tau}}$ such that

$$\lim_{t\to\infty} \boldsymbol{\eta}(t) - \boldsymbol{\eta}_c(t) = 0 \quad or \quad \lim_{t\to\infty} \boldsymbol{\eta}_e(t) = 0 \tag{3.10}$$

where $\boldsymbol{\eta}_c(t)$ is the velocity controller designed in the first step and $\boldsymbol{\eta}_e(t)$ is the joint velocities error between actual and controlled joint velocities.

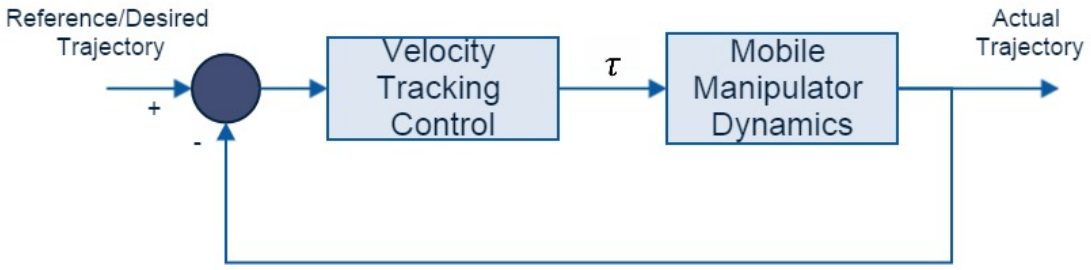


Figure 3.1: Block diagram of control scheme

## 3.2 Control Design

### 3.2.1 Kinematic Controller

We will use the backstepping approach to design a kinematic or velocity controller. To design a motion tracking controller, we only consider the Eq.(2.35) which is the kinematic equation of the mobile manipulator.

$$\dot{\mathbf{q}} = H(q)\boldsymbol{\eta}. \tag{3.11}$$

Firstly define an output of the system as

$$\boldsymbol{\Psi} = f(q), \tag{3.12}$$

The error and its derivative can be written as

$$\begin{aligned} \mathbf{e} &= \boldsymbol{\Psi}_d - \boldsymbol{\Psi} \\ \dot{\mathbf{e}} &= \dot{\boldsymbol{\Psi}}_d - \dot{\boldsymbol{\Psi}} \end{aligned}, \tag{3.13}$$

where $\mathbf{e}$ is the velocity error of the mobile platform and the position error of the end effector. Taking the time derivative of the Eq.(3.12)

$$\dot{\boldsymbol{\Psi}} = \frac{\partial f(q)}{\partial q}\dot{\mathbf{q}} = \frac{\partial f(q)}{\partial q}H(q)\boldsymbol{\eta} = \Delta\boldsymbol{\eta}, \tag{3.14}$$

where $\Delta$ is a decoupling matrix or extended Jacobian matrix and can be represented as

$$\Delta = \frac{\partial f(q)}{\partial q}H(\mathbf{q}). \tag{3.15}$$

From Eqs. (3.13) and (3.14),

$$\dot{\mathbf{e}} = \dot{\boldsymbol{\Psi}}_d - \Delta\boldsymbol{\eta}. \tag{3.16}$$

Let $\boldsymbol{\eta} = \boldsymbol{\eta}_c$, as the virtual velocity control,Eq.(3.16) can be written as,

$$\dot{\mathbf{e}} = \dot{\boldsymbol{\Psi}}_d - \Delta\boldsymbol{\eta}_c. \tag{3.17}$$

Choosing the Lyapunov function as,

$$V_1 = \frac{1}{2}\mathbf{e}^T\mathbf{e}. \tag{3.18}$$

Taking the derivative gives,

$$\begin{aligned} \dot{V}_1 &= \mathbf{e}^T\dot{\mathbf{e}} \\ &= \mathbf{e}^T(\dot{\boldsymbol{\Psi}}_d - \Delta\boldsymbol{\eta}_c) \end{aligned} \tag{3.19}$$

Choosing the kinematic control law $\boldsymbol{\eta}_c$

$$\eta_c = \Delta^{-1}[\dot{\boldsymbol{\Psi}}_d + K\mathbf{e}], \tag{3.20}$$

where $K > 0$ is the positive constant control gain. Eq.(3.20) is the velocity tracking kinematic controller. From Eqs.(3.17) and (3.20)

$$\dot{\mathbf{e}} = \dot{\boldsymbol{\Psi}}_d - \Delta[\Delta^{-1}(\dot{\boldsymbol{\Psi}}_d + K\mathbf{e})] = -K\mathbf{e}. \tag{3.21}$$

From Eq.(3.21) when $K > 0$, $\lim_{t\to\infty}\mathbf{e}(t) = 0$.

$$\dot{V}_1 \leq -K||\mathbf{e}||^2.$$

If $\Delta$ is a full rank square matrix, then $\Delta^{-1}$ can be obtained via simple inversion. If $\Delta$ is not a full rank matrix or the mobile manipulator is at a singular configuration, the system given in Eq.(3.11) contains linearly dependent equations. In this case, $\Delta^{-1}$ cannot be derived from the simple inversion of $\Delta$. An alternative solution to invert the $\Delta$ at the sinuglar point or in the neighbourhood of a singularity is provided by the DLS inverse method [49, 50]. In the Damped Least Square inverse method, the inverse can be written by Eq.(3.22)

$$\Delta^{-1} = \Delta^T\Big(\Delta\Delta^T + k^2I\Big)^{-1}, \tag{3.22}$$

where $k$ is a damping factor.

### 3.2.2 Adaptive Torque Control

In the first part of the controller design, kinematic velocity controller is designed to track the desired trajectory of the mobile manipulator system. Using the controller $\boldsymbol{\eta}_c$, the adaptive torque controller is designed. System given in Eq.(2.37) is considered to design the controller. Using the properties of dynamic system given in Eqs.(2.40), (2.41) and (2.42), the adaptive torque controller is designed as follows.

Consider a Lyapunov function $V_2$ as

$$V_2 = \frac{1}{2}\boldsymbol{\eta}_e^T \bar{M} \boldsymbol{\eta}_e, \tag{3.23}$$

where $\boldsymbol{\eta}_e$ is defined as a state error of the kinematically controlled system and can be written as Eq.(3.24). Note that it is not an independent Lyapunov function for the adaptive torque control as $\eta_e$ given in the Eq.(3.24) depends upon the kinematic control,

$$\boldsymbol{\eta}_e = \boldsymbol{\eta} - \boldsymbol{\eta}_c. \tag{3.24}$$

Taking the time derivative of the Lyapunov function $V_2$ as

$$\dot{V}_2 = \frac{1}{2}\boldsymbol{\eta}_e^T \Big[\dot{\bar{M}}\boldsymbol{\eta}_e + 2\bar{M}\dot{\boldsymbol{\eta}}_e\Big]. \tag{3.25}$$

From Eq.(3.24)

$$\dot{V}_2 = \boldsymbol{\eta}_e^T \Big[\frac{1}{2}\dot{\bar{M}}\Big(\boldsymbol{\eta} - \boldsymbol{\eta}_c\Big) + \bar{M}\Big(\dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_c\Big)\Big]. \tag{3.26}$$

From Eq.(3.1), we have

$$\begin{aligned}
\dot{V}_2 &= \boldsymbol{\eta}_e^T \left[\frac{1}{2}\dot{\bar{M}}\Big(\boldsymbol{\eta} - \boldsymbol{\eta}_c\Big) + \bar{M}\Big[\bar{M}^{-1}\Big(\bar{\boldsymbol{\tau}} - \bar{C}\boldsymbol{\eta} - \bar{G}\Big) - \dot{\boldsymbol{\eta}}_c\Big]\right] \\
&= \boldsymbol{\eta}_e^T \left[\frac{1}{2}\dot{\bar{M}}\Big(\boldsymbol{\eta} - \boldsymbol{\eta}_c\Big) + \bar{\boldsymbol{\tau}} - \bar{C}\boldsymbol{\eta} - \bar{G} - \bar{M}\dot{\boldsymbol{\eta}}_c\right] \\
&= \boldsymbol{\eta}_e^T \left[\frac{1}{2}\dot{\bar{M}}\boldsymbol{\eta} - \frac{1}{2}\dot{\bar{M}}\boldsymbol{\eta}_c + \bar{\boldsymbol{\tau}} - \bar{C}\boldsymbol{\eta} - \bar{G} - \bar{M}\dot{\boldsymbol{\eta}}_c\right].
\end{aligned} \tag{3.27}$$

Adding and subtracting $\bar{C}\eta_C$ in Eq.(3.27) and re-arranging the terms, we obtain

$$\dot{V}_2 = \boldsymbol{\eta}_e^T \left[\Big(\frac{1}{2}\dot{\bar{M}}\boldsymbol{\eta} - \bar{C}\boldsymbol{\eta}\Big) - \Big(\frac{1}{2}\dot{\bar{M}}\boldsymbol{\eta}_c - \bar{C}\boldsymbol{\eta}_c\Big) - \bar{C}\boldsymbol{\eta}_c + \bar{\boldsymbol{\tau}} - \bar{G} - \bar{M}\dot{\boldsymbol{\eta}}_c\right]. \tag{3.28}$$

Using the dynamic property of the mobile manipulator system, from Eq.(2.41) we can simplify the Eq.(3.28) as follow

$$\dot{V}_2 = \boldsymbol{\eta}_e^T \Big[\bar{\boldsymbol{\tau}} - \bar{M}\dot{\boldsymbol{\eta}}_c - \bar{C}\boldsymbol{\eta}_c - \bar{G}\Big]. \tag{3.29}$$

Using the dynamic property in Eq.(2.42), the derivative of the Lyapunov function can be written as

$$\dot{V}_2 = \boldsymbol{\eta}_e^T \Big[ \bar{\boldsymbol{\tau}} - Y\Big(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\eta}_c, \dot{\boldsymbol{\eta}}_c\Big)\mathbf{p} \Big]. \tag{3.30}$$

Therefore, the control law for the known parameters $\mathbf{p}$ can be chosen as,

$$\bar{\boldsymbol{\tau}} = Y\Big(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\eta}_c, \dot{\boldsymbol{\eta}}_c\Big)\mathbf{p} - K_1 \boldsymbol{\eta}_e, \tag{3.31}$$

where $K_1$ is a constant control gain positive definite matrix.
With $K_1$ being positive definite matrix,

$$\dot{V}_2 = -\boldsymbol{\eta}_e^T K_1 \boldsymbol{\eta}_e \leq -\lambda_{min}(K_1)||\boldsymbol{\eta}_e||^2. \tag{3.32}$$

However here in Eq.(3.31), $\mathbf{p}$ is the vector of uncertain or unknown parameters. So we need to add the adaptive controller to estimate the constant value of vector $\mathbf{p}$. Define the estimated parameter vector is $\hat{\mathbf{p}}$ and then define the error between actual and estimated parameters as

$$\mathbf{e_p} = \mathbf{p} - \hat{\mathbf{p}}. \tag{3.33}$$

Choosing the control law $\bar{\boldsymbol{\tau}}$, and parameter update law $\dot{\hat{\mathbf{p}}}$ as

$$\begin{aligned} \bar{\boldsymbol{\tau}} &= Y(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\eta}_c, \dot{\boldsymbol{\eta}}_c)\hat{\mathbf{p}} - K_1 \boldsymbol{\eta}_e \\ \dot{\hat{\mathbf{p}}} &= -(\Gamma^{-1})^T Y(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\eta}_c, \dot{\boldsymbol{\eta}}_c)^T \boldsymbol{\eta}_e \\ &= -\Gamma^{-1} Y(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\eta}_c, \dot{\boldsymbol{\eta}}_c)^T \boldsymbol{\eta}_e \end{aligned} \tag{3.34}$$

where $\Gamma$ is the symmetric positive definite gain matrix which ensures that Eq.(3.35) is positive definite.
Consider the Lyapunov function $V_3$ as

$$V_3 = V_2 + \frac{1}{2}\mathbf{e}_p^T \Gamma \mathbf{e}_p. \tag{3.35}$$

Taking the time derivative of the Lyapunov function,

$$\begin{aligned} \dot{V}_3 &= \boldsymbol{\eta}_e^T \Big[ \bar{\boldsymbol{\tau}} - Y(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\eta}_c, \dot{\boldsymbol{\eta}}_c)\mathbf{p} \Big] + \dot{\mathbf{e}}_p{}^T \Gamma \mathbf{e}_p \\ &= \boldsymbol{\eta}_e^T \Big[ \bar{\boldsymbol{\tau}} - Y\mathbf{p} \Big] + (-\dot{\hat{\mathbf{p}}})^T \Gamma (\mathbf{p} - \hat{\mathbf{p}}) \end{aligned} \tag{3.36}$$

Using the parameter update law from Eq.(3.34),

$$\begin{aligned}
\dot{V}_3 &= \boldsymbol{\eta}_e^T \Big[\bar{\boldsymbol{\tau}} - Y\mathbf{p}\Big] + (\boldsymbol{\eta}_e^T) Y (\Gamma^{-1}\Gamma)(\mathbf{p} - \hat{\mathbf{p}}) \\
&= \boldsymbol{\eta}_e^T \Big[\bar{\boldsymbol{\tau}} - Y\mathbf{p} + Y\mathbf{p} - Y\hat{\mathbf{p}}\Big] \\
&= \boldsymbol{\eta}_e^T \Big[\bar{\boldsymbol{\tau}} - Y\hat{\mathbf{p}}\Big] \\
&\leq \boldsymbol{\eta}_e^T K_1 \boldsymbol{\eta}_e \\
&\leq -\lambda_{min}(K_1)||\boldsymbol{\eta}_e||^2. \qquad (3.37)
\end{aligned}$$

From the Lyapunov stability theorem and definition given in Eqs.(3.3)-(3.6), $\boldsymbol{\eta}_e \in L_\infty$. Integrating Eq.(3.37) we get,

$$V_2(t) \leq V_2(0) - \lambda_{min} K_1 \int_0^t ||\boldsymbol{\eta}_e||^2 dt.$$

Rearranging the above equation,

$$\int_0^t ||\boldsymbol{\eta}_e||^2 dt \leq \frac{V_2(0) - V_2(t)}{\lambda_{min} K_1},$$

Since $V_2(0)$ is finite and $V_2(t) \in L_\infty$, we can write, $\boldsymbol{\eta}_e \in L_2$. Also, because $\dot{\boldsymbol{\eta}}_e$ is a function of $\boldsymbol{\eta}_e$. In Eqs.(3.32) and (3.37), $\lambda_{min}(K_1)$ is the minimum eigenvalue of the control gain $K_1$. Thus from the Barbalat's lemma we can write

$$\lim_{t\to\infty} \boldsymbol{\eta}_e = 0 \Longrightarrow \lim_{t\to\infty} \boldsymbol{\eta} = \boldsymbol{\eta}_c. \qquad (3.38)$$

Hence the system tracking error tends to zero asymptotically and the system in (3.34) is Lyapunov stable.

# Chapter 4

## Simulation Results on Model Based Control

The chapter descries the model based control of mobile manipulators, with the aid of the concept of feedback linearization. The basic idea of utilizing this approach in the chapter is to assist in the conversion of the nonlinear system into a full or partial linear system, by algebraic calculations so as to apply linear control techniques. This is applied to holonomic constrained nonholonomic mobile manipulators. In turn, the chapter outlines the attempt to provide a solution to motion/force control, citing limitations of the model in the sensitivity to the accuracy of the dynamic model. Simulation was performed to determine the effectiveness of the proposed controller. MATLAB/Simulink was used for the simulation purpose. Simulink is a MATLAB toolbox, which provides customized block libraries and solvers for modeling and simulating dynamic systems. The simulation procedure in Simulink can be described by Fig.4.1.

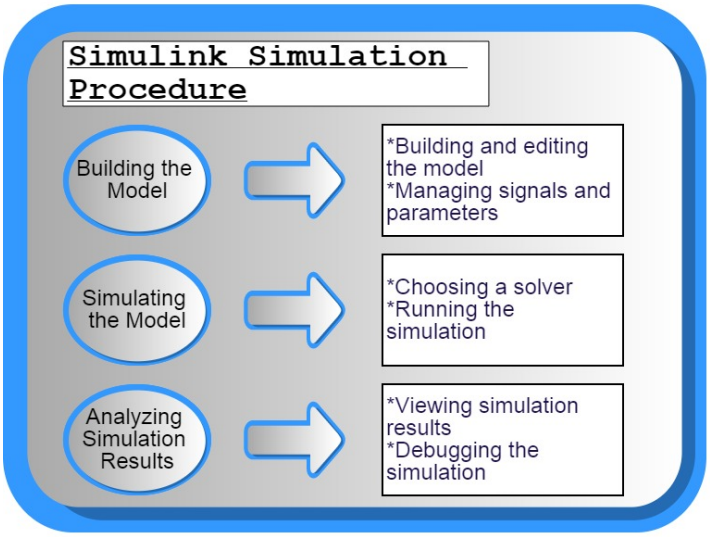


Figure 4.1: Simulation procedure in MATLAB/Simulink

## 4.1 Model Based Control (MBC)

A complicated dynamic model of the mobile manipulator is created in the Simulink by programming the MATLAB Function Block. Case study on a model based controller was performed in Simulink. The model based control architecture [4] implemented in the Simulink. The generalized block diagram of the model based control is shown in Fig.4.4. The closed loop system was modeled in Simulink for the holonomic constrained nonholonomic mobile manipulator. The overall Simulink block diagram is shown in the Fig.4.2.

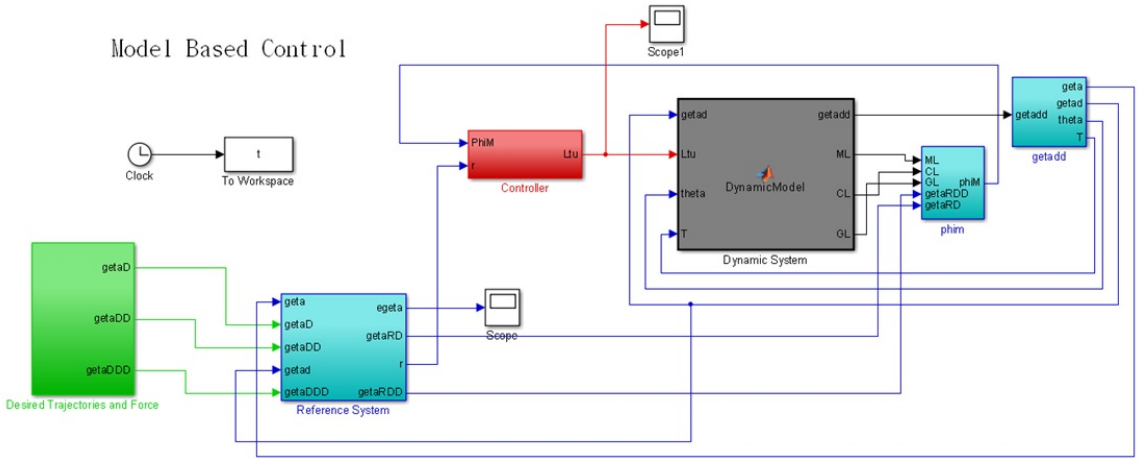


Figure 4.2: Simulation block diagram of model based controller

The mobile manipulator assumed was subjected to the holonomic and nonholonomic constraints. The mobile manipulator was subjected to the following constraints given in Eq.(4.1).

$$\begin{aligned} -\dot{x}\sin\theta + \dot{y}\cos\theta + l\dot{\theta} + r\theta_L &= 0 \\ \dot{x}\sin\theta - \dot{y}\cos\theta + l\dot{\theta} + r\theta_R &= 0 \\ \dot{x}\cos\theta + \dot{y}\sin\theta &= 0. \end{aligned} \tag{4.1}$$

The mobile manipulator system considered for this simulation is shown in Fig.4.3. The manipulator is mounted on the nonholonomic mobile base such that the distance between the center of the wheel and the first link of manipulator is $d$ as shown in the figure. The reduced dynamic model of a mobile manipulator was considered from the

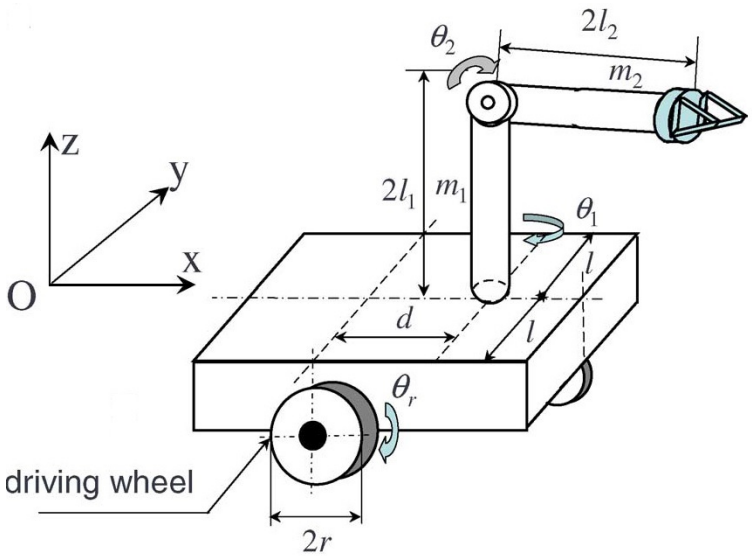


Figure 4.3: Mobile manipulator system considered for the model based control [4]

reference [4] given in Eqs.(4.2) and (4.1) and details of the dynamic terms are given in Appendix A.4.

$$M_L\ddot{\boldsymbol{\zeta}} + C_L\dot{\boldsymbol{\zeta}} + G_L + d_L(t) = L^T\mathbf{u} \tag{4.2}$$

where

$$\begin{aligned} M_L &= L^T(\zeta)M^1L \\ C_L &= L^T(\zeta)C^1 \\ G_L &= L^T(\zeta)G^1 \\ d_L &= L^T(\zeta)d^1(t). \end{aligned}$$

Due to the space limit only reduced dynamic equations are shown here. The detailed dynamics and its reduction is given in the literature [4]. The system has several dynamic properties same as described in the dynamic properties section. Mobile manipulator dynamics are as in Eqs.(2.40), (2.41) and (2.42).

**Assumption 1**

The desired trajectory $\zeta_d(t)$ is assumed to be uniformly bounded and continuous. It is uniformly bounded and the second order derivative is continuous. This assumption is made because it is required for the controller design where $\dot{\zeta}_d(t)$ is used as in Eq.(4.3),

$$\begin{aligned} \mathbf{e}_\zeta &= \boldsymbol{\zeta} - \boldsymbol{\zeta}_d \\ \dot{\boldsymbol{\zeta}}_r &= \dot{\boldsymbol{\zeta}}_d - K_\zeta\mathbf{e}_\zeta \\ \mathbf{r} &= \dot{\mathbf{e}}_\zeta + K_\zeta\mathbf{e}_\zeta \end{aligned} \tag{4.3}$$

The system dynamics can be written in the following form,

$$M_L\ddot{\boldsymbol{\zeta}} + C_L\dot{\boldsymbol{\zeta}} + G_L + d_L = L^T\mathbf{u}_a. \tag{4.4}$$

Considering there is no external disturbance, i.e. $d_L = 0$, and the dynamics of mobile manipulator is known completely, the model based control law used for the simulation is given by Eq.(4.5),

$$L^T u_a = -K_p r - K_i \int_0^t r ds - \Phi_m \tag{4.5}$$

where

$$\begin{aligned}
\Phi_m &= C^m \Psi_m \\
C_m &= [M_L \quad C_L \quad G_L] \\
\Psi_m &= [\ddot{\zeta}_r \quad \dot{\zeta}_r \quad 1]^T
\end{aligned}$$

Table 4.1: Mobile manipulator parameters used in simulation for the MBC control [4].

| Parametrs | Values | Units |
|---|---|---|
| Mass of the mobile base ($m$) | 1.0 | $kg$ |
| Mass of Link 1 ($m_1$) | 1.0 | $kg$ |
| Mass of Link 2 ($m_2$) | 1.0 | $kg$ |
| Moment of inertia of the mobile base ($I_p$) | 0.5 | $kgm^2$ |
| Moment of inertia of Link 1 ($I_1$) | 1.0 | $kgm^2$ |
| Moment of inertia of Link 2 ($I_2$) | 1.0 | $kgm^2$ |
| Wheel radius ($R$) | 1.0 | $m$ |
| Length of Link 1 ($l_1$) | 0.5 | $m$ |
| Length of Link 2 ($l_1$) | 0.5 | $m$ |
| d | 1.0 | $m$ |
| Initial $q(0)$ | $[0, 4, 0.785, 0.1]^T$ | $[m, m, rad, rad]^T$ |
| Gravitational force ($g$) | 9.8 | $m/s^2$ |

Table 4.2: Control gains used for the model based control

| Control Gain | Value |
|---|---|
| $K_p$ | $Diag[1.0]$ |
| $K_\zeta$ | $Diag[1.0]$ |
| $K_i$ | 0 |

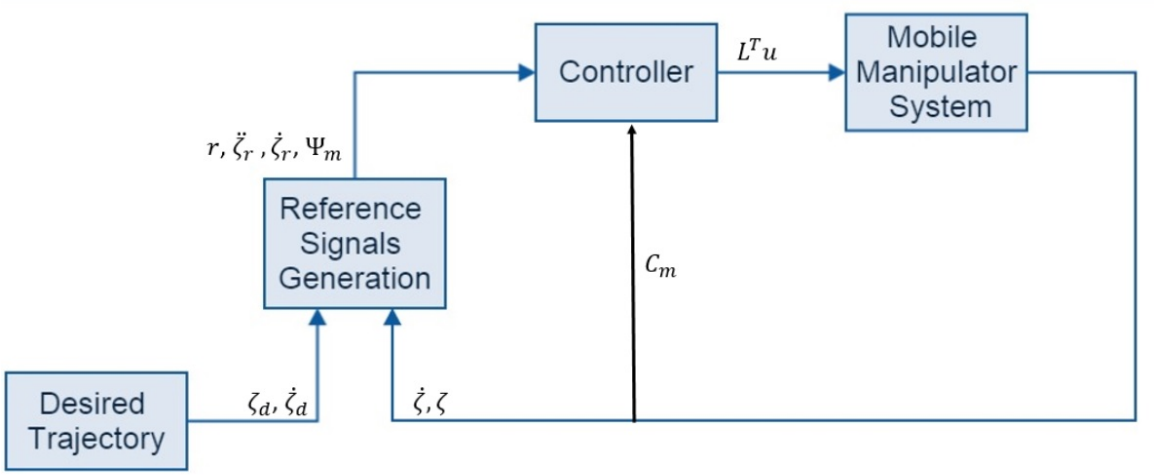


Figure 4.4: Generalized block diagram of model based control

## 4.2 Simulation Results

The parameters of the mobile manipulator used to test the model based control are shown in Tables 4.1 and 4.2.
The desired trajectories are given as

$$y_d = 1.5\sin(t)$$
$$\theta_d = 1.0\sin(t)$$
$$\theta_{1d} = \frac{\pi}{4(1-\cos(t))}$$

Under the assumptions that the system model is completely known, the model based control used to track the desired motion trajectories over the specified time period. The trajectory tracking of the model based controller is shown in Fig.4.6.

From the figures, it is clear that the model-based control achieves the desired responses. In Fig.4.5, individual joint positions tracking is shown. Actual responses track the desired responses with some tracking error as shown in Fig.4.6. In Fig.4.7, the error signals are not smooth and they are fluctuating throughout. The tracking error results are not satisfactory for the given case. Also, the torque changing more

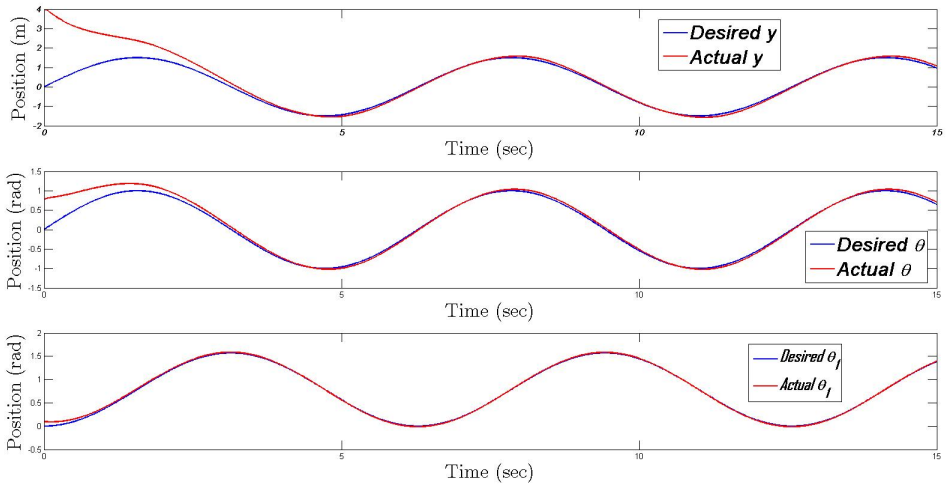


Figure 4.5: Individual joint position tracking by model-based control.

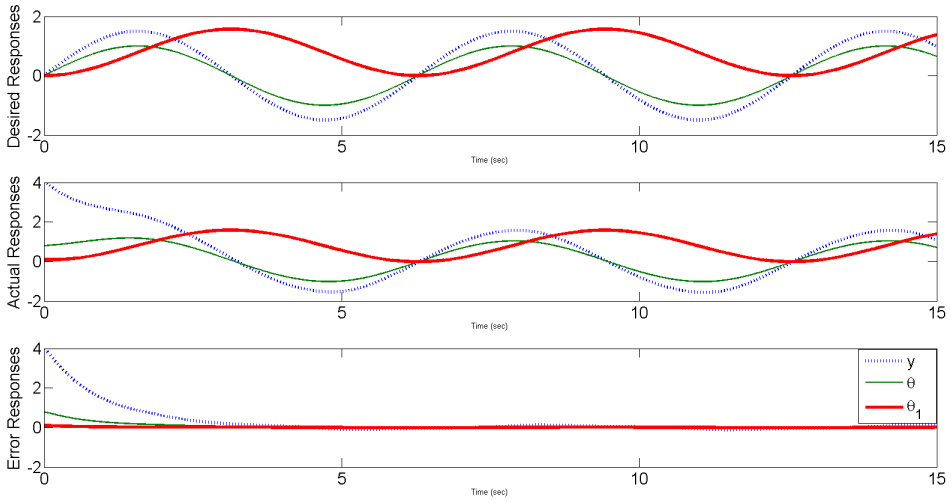


Figure 4.6: Joint position trajectory tracking by model-based control.

greatly for the model-based control as shown in Fig.4.8. Although the control input is bounded, there is a high fluctuation in the torque throughout the simulation procedure. The presented system doesn't consider any disturbances and it doesn't have any singularity in the system. The responses can be smoothly bounded by applying a backstepping-adaptive controller for the mobile manipulator system subjected to singularity. The velocities of the joints with MBC are compared with the backstepping-adaptive control in the next chapter.

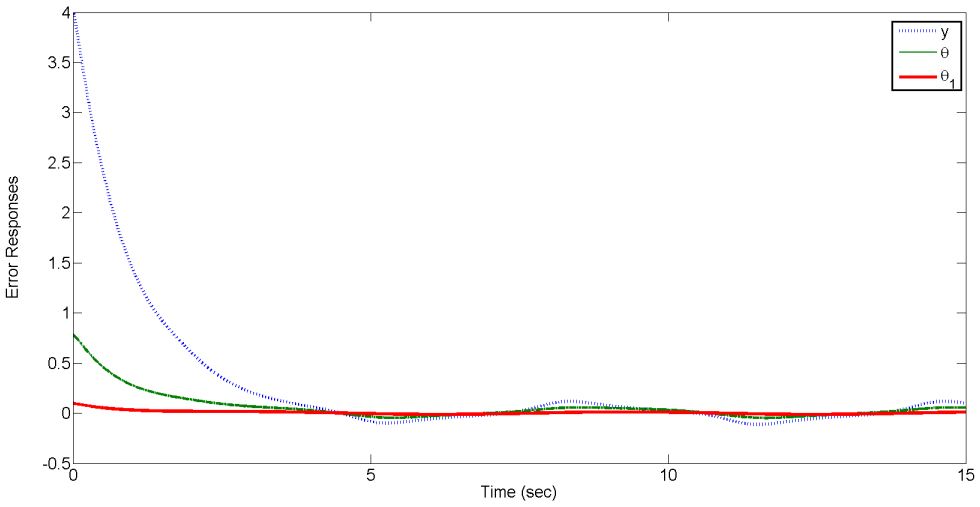


Figure 4.7: Trajectory tracking error by model-based control.

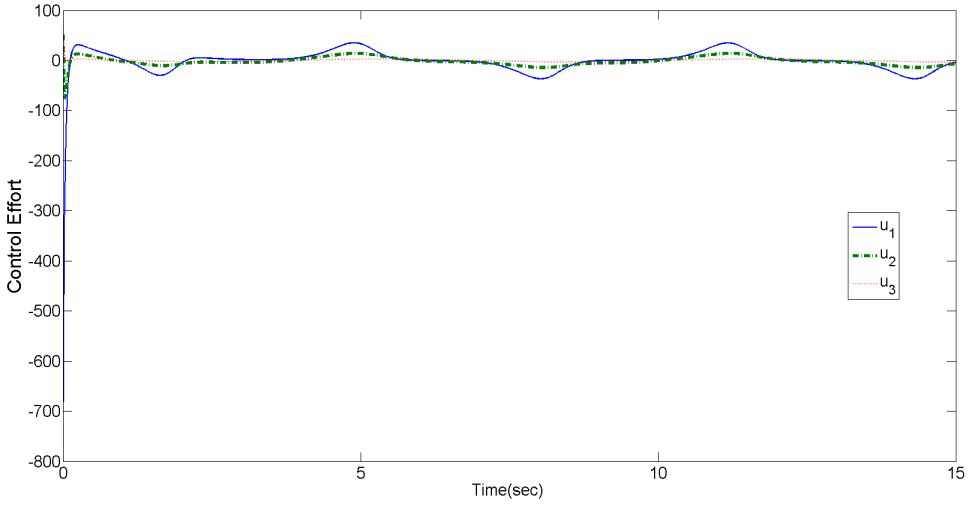


Figure 4.8: Torque of the joints by model-based control.

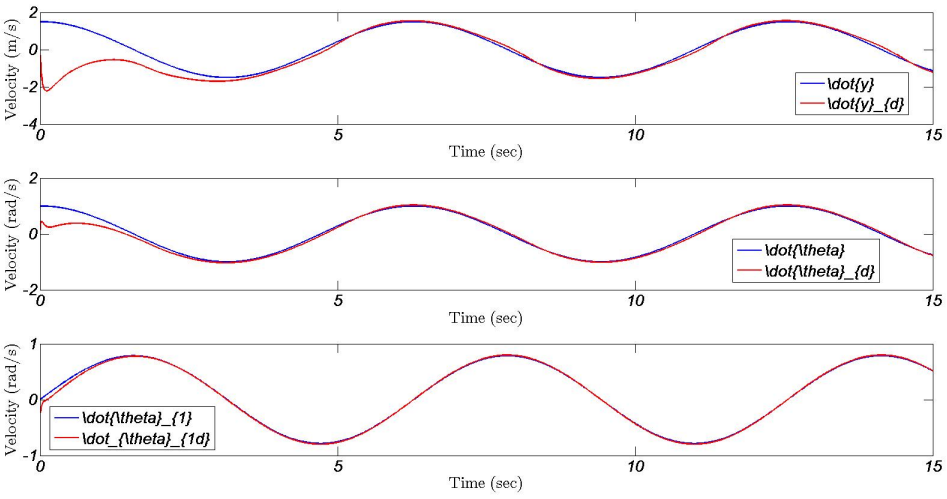


Figure 4.9: Velocities of the joints by model based control

# Chapter 5

# Simulation Results on Backstepping Adaptive Control

## 5.1 System Model and Control Scheme in MATLAB/Simulink

In Section 5.1, simulation results of the mobile manipulator are presented shown in the Fig.5.1 using the controller presented in Eq.(3.34) for the system presented in Eq.(3.1). The proposed controller is tested for the trajectory tracking of the end-effector of the mobile manipulator along with the velocity tracking of the mobile base.

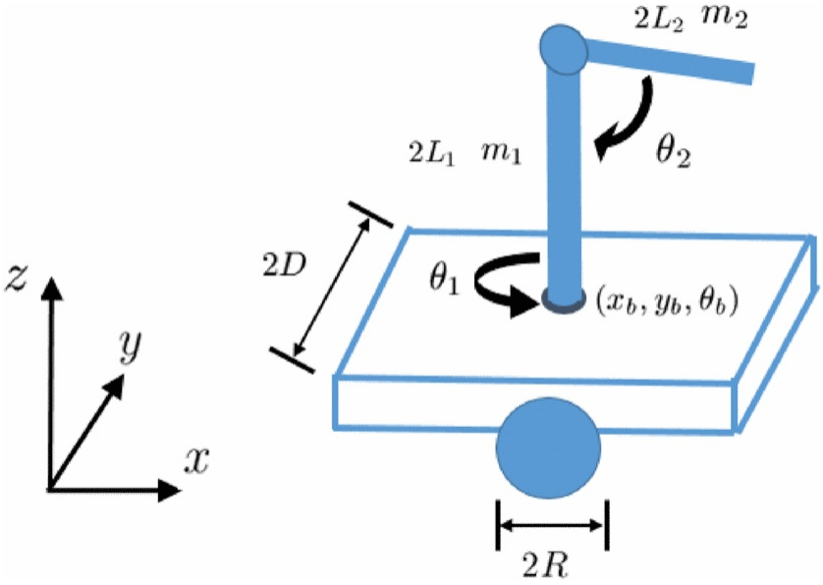


Figure 5.1: Mobile manipulator considered in the simulation [5]

First consider a nonholonomic wheeled mobile manipulator shown in Fig.5.1. The definitions of the parameters of the mobile manipulator are presented here in Table 5.1. The dynamics of the mobile manipulator system is given in Eqs.(2.21)-(2.22) and (2.35)-(2.39). The detailed dynamics of the mobile manipulator is presented in Appendix A.4 . All the values of the parameters of the mobile manipulator considered for the simulation are shown in Table 5.2.

Table 5.1: Mobile manipulator variables

| variable | Description |
|---|---|
| $\tau_r$, $\tau_l$ | The torque of the two wheels |
| $\theta_r$, $\theta_l$ | The rotation angle of left and right wheel of the mobile manipulator |
| $v$ | Forward velocity of the mobile manipulator platform |
| $w$ | Rotation velocity of the mobile manipulator platform |
| $m_1$, $m_2$ | the mass of link 1 and 2 |
| $m_p$ | the mass of the mobile manipulator platform |
| $2L_1$, $2L_2$ | the length of link 1 and 2 |
| $R$ | the radius of the wheel of the mobile manipulator |
| $x$, $y$, $\theta_b$ | Position and orientation of the mobile manipulator platform |
| $2D$ | Distance between the wheels |
| $\theta_1$, $\theta_2$ | Angular position of link 1 and 2 |
| $I_b$ | Moment of inertia of the mobile manipulator platform |
| $I_1$, $I_2$ | Moment of inertia of link 1 and 2 |
| $g$ | gravitational force |

Table 5.2: Mobile manipulator parameters used in simulation for proposed control

| Parametrs | Values | Units |
|---|---|---|
| $m_p$ | 32 | $kg$ |
| $m_1$ | 3 | $kg$ |
| $m_2$ | 3 | $kg$ |
| $I_b$ | 1 | $kgm^2$ |
| $I_1$ | 1 | $kgm^2$ |
| $I_2$ | 1 | $kgm^2$ |
| $R$ | 0.5 | $m$ |
| $L_1$ | 1 | $m$ |
| $L_2$ | 1 | $m$ |
| $g$ | 9.8 | $m/s^2$ |

The controller implemented for the control of the mobile manipulator can be described in Fig.5.2. The same scheme was modelled in the Simulink for a given mobile manipulator system to track the given desired trajectories $\Psi_d$.

Fig.5.3 shows the overall closed-loop system of the mobile manipulator with the proposed controller in the Simulink environment. The nonholonomic dynamic model of the 2-link mobile manipulator system described in Eq.(3.1) is programmed in the "Mobile Manipulator System" block. The details of the DLS inverse method given in Eq.(3.22) are programmed in "Kinematic Transformation and DLS" block. The

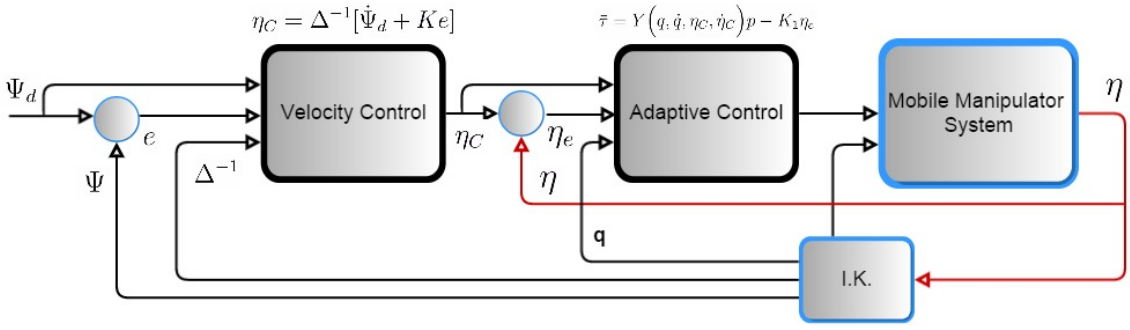


Figure 5.2: Backstepping-Adaptive Control Scheme

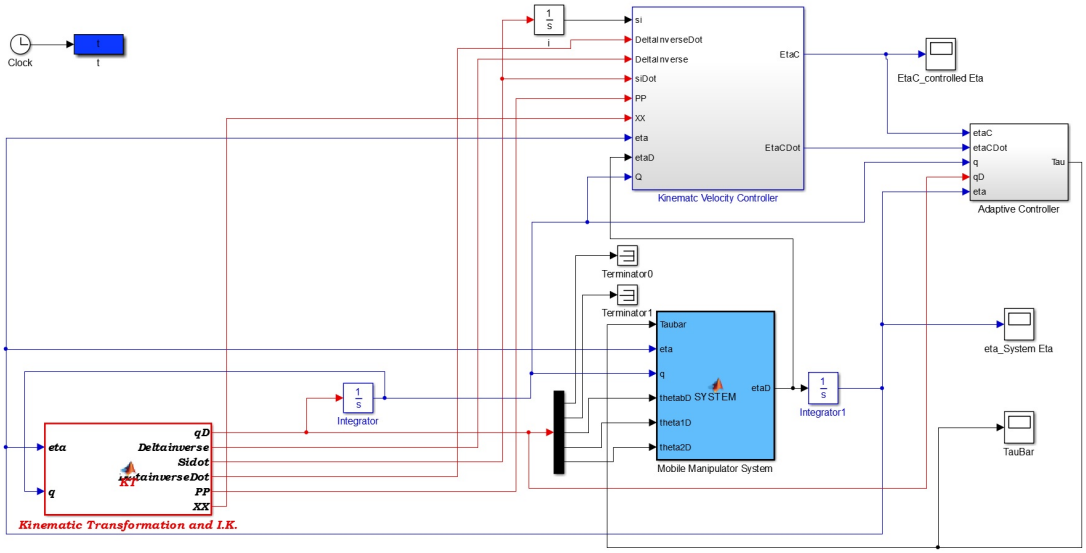


Figure 5.3: Simulink block diagram

backstepping kinematic control of mobile manipulator is constructed in the "Kinematic Velocity Controller" subsystem and the adaptive torque control is modelled in the "Adaptive controller" subsystem. The "ode45(Dorman-Prince)" solver is used to run the simulation over the 10 seconds of time to analyse the performance of the proposed control to track the different given desired trajectories.

For the following part of the chapter the mobile manipulator system and its parameters will be same as discussed unless specified. Three cases are considered. The first and second cases while mobile manipulator system's output operational space has five states $\mathbf{\Psi}_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}, z_{EE_d}\right]^T$; and the third case when mobile manipulator system's output operational space has four states $\mathbf{\Psi}_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}\right]^T$. In the third case, since the operational space has four vectors and the Jacobian matrix is

$4 \times 4$, the Jacobian inverse of the kinematics can be calculated directly by inverting the matrix using the matrix laws. In the first and second cases, the operational space has five vectors and because of the extra state in the operational space, the size of the Jacobian matrix is $5 \times 4$, which cannot be inversed using the same law. DLS inverse method is used to invert the Jacobian and remove the singularity from the system. It gives the approximation of the Jacobian inverse. For the first and second cases, different trajectories of the end-effector is tracked with the proposed controller. For the third case, the same trajectory used for Case 1 is tracked to verify the controller. However the control gain values are not the same in both cases. Note that in both cases, the mathematical models of the system are the same. However the inverse kinematics of the mobile manipulator system are different in both cases i.e. Case 1 and Case 3. Because in Case 1, the angular velocity of the wheels of the mobile platform is used to calculate the inverse kinematics states. In Case 3, the mobile platform velocity and acceleration is used as the inverse kinematics states. For both cases they have multiple solutions for their inverse kinematics because of the redundancy. A redundancy resolution method is used to remove the singularity from the system. Defining 3 cases for the simulation process as follows,

1. Trajectory tracking Case 1: The system's output operational space has five states $\boldsymbol{\Psi}_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}, z_{EE_d}\right]^T$ to track the trajectory given in Eq.(5.1).

2. Trajectory tracking Case 2: The system's output operational space has five states $\boldsymbol{\Psi}_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}, z_{EE_d}\right]^T$ to track the trajectory given in Eq.(5.2).

3. Trajectory tracking Case 3: The system's output operational space has four states $\boldsymbol{\Psi}_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}\right]^T$ to track the trajectory given in Eq.(5.1).

## 5.2 Desired Trajectory Tracking of the End-Effector Position and Vehicle Velocity

In this section, we assumed that the output of the mobile manipulator system has five states $\boldsymbol{\Psi}_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}, z_{EE_d}\right]^T$. This means that the Jacobian matrix is not a square matrix and it cannot be inverted using the direct inversion of the matrix. For that, we used the DLS inverse method. In the following section, the simulation of

the mobile manipulator system for the trajectory tracking of the mobile manipulator is given. The proposed controller is used to verify the effectiveness of the modeling and the control of the mobile manipulator system.

### 5.2.1 Trajectory Tracking Case 1

Consider the given mobile manipulator system, the time dependent desired trajectory $\Psi_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}, z_{EE_d}\right]^T$ is selected as Eq.(5.1).

$$\begin{cases} \dot{x} & = 0.1 + 0.05\sin t \\ \dot{y} & = 0.1 + 0.05\cos t \\ x_{EE_d} & = 0.1t - 0.05\cos t + 0.05 + 1 \\ y_{EE_d} & = 0.1t + 0.05\sin t + 1 \\ z_{EE_d} & = 2 \end{cases} \tag{5.1}$$

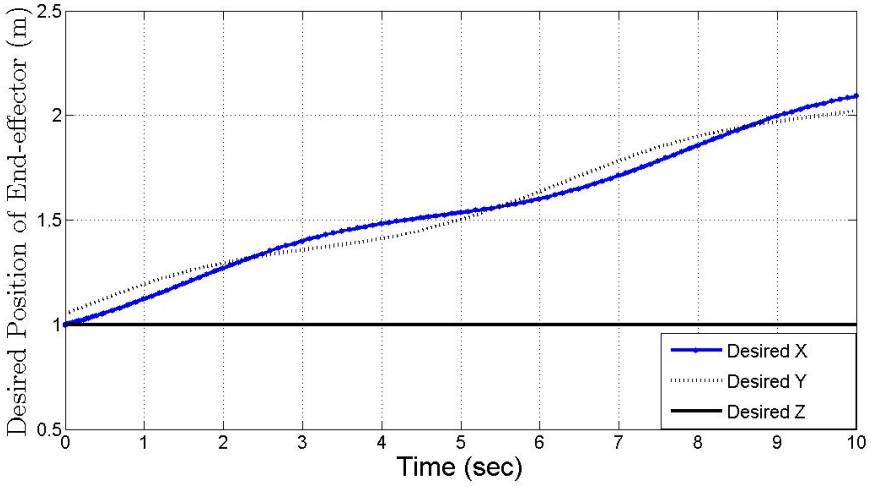


Figure 5.4: Desired end-effector trajectories for Case 1

The desired trajectory of the end-effector $\Psi_{ad}(t) = \left[x_{EE_d}, y_{EE_d}, z_{EE_d}\right]^T$ is shown in Fig.5.4, where desired $X$, $Y$ and $Z$ refers to the desired $X, Y$ and $Z$ positions of the end-effector which are $x_{EE_d}, y_{EE_d}$ and $z_{EE_d}$. Note that the desired $Z$ position of the end-effector is chosen as a constant. The initial position of the end-effector is given in Table 5.3.

The control gain values chosen for this case are given in Table 5.4.

Table 5.3: Initial position of the end-effector of the mobile manipulator for case 1

| State | Initial Position |
|---|---|
| $\dot{x}$ | 0 |
| $\dot{y}$ | 0 |
| $x_{EE_d}$ | 0 |
| $y_{EE_d}$ | 1 |
| $z_{EE_d}$ | 2 |

Table 5.4: Control gain values for case 1

| Control Gain | Value |
|---|---|
| $k$ | 1.01 |
| $K$ | 2.6 |
| $K_1$ | $80I_{4\times4}$ |
| $\Gamma$ | $-10I_{6\times6}$ |

The main objective is to track the desired trajectory with minimum possible errors. Results are plotted and shown in the following section to analyze the performance of the controller. Actual $X$, $Y$ and $Z$ positions of the manipulator end-effector are shown in Figs.5.5, 5.6 and 5.7. These figures clearly show that the actual position of the end-effector is tracking the desired positions of the end-effector given in Fig.5.4.

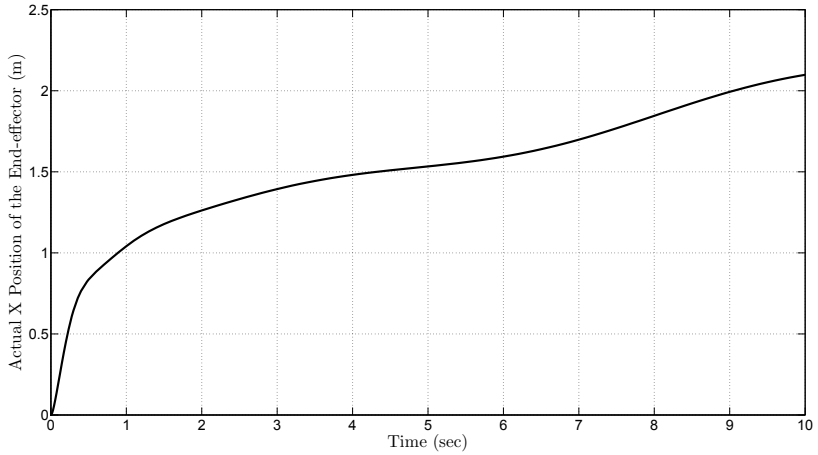


Figure 5.5: Actual $x_{EE}$ of Case 1

Figs.5.8, 5.9 and 5.10 describe the desired and actual end-effector position of mobile manipulator. The actual position of the end-effector converge to the desired

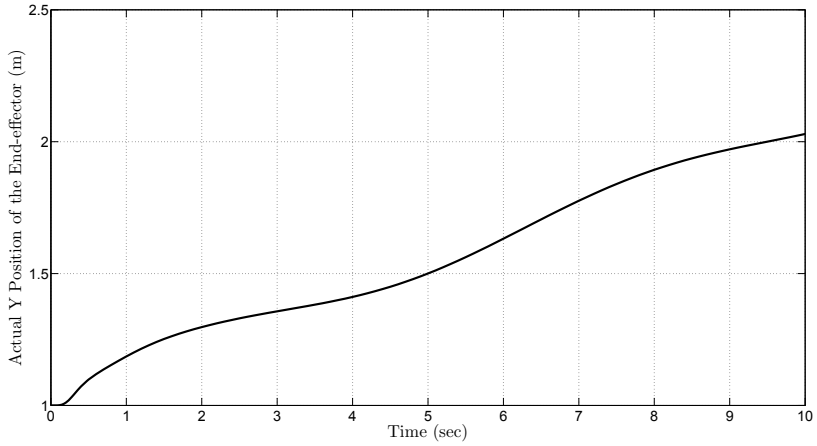


Figure 5.6: Actual $y_{EE}$ of Case 1

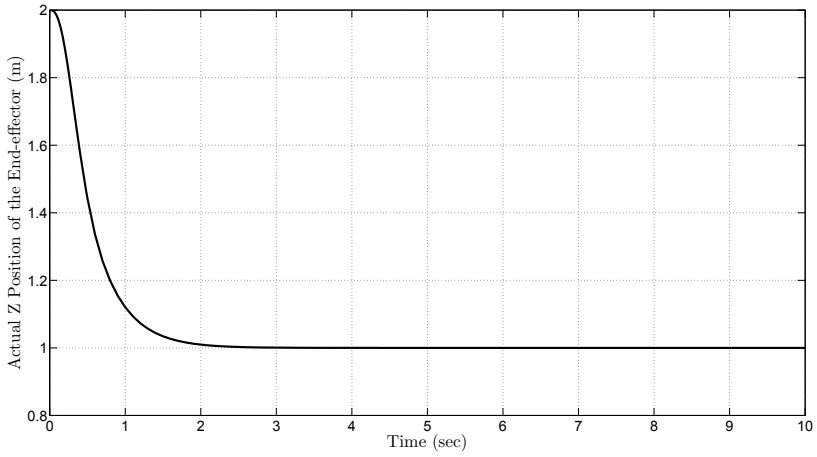


Figure 5.7: Actual $z_{EE}$ of Case 1

trajectory quickly with zero error. These results show the perfect trajectory tracking of the end-effector and the effectiveness of the proposed controller for the given redundant mobile manipulator system. The convergence time of the actual trajectory to desired trajectory is around 2 seconds. The errors between the desired and the actual position are shown in Fig.5.11.

From Fig.5.11, the position errors of $X$, $Y$ and $Z$ of the mobile manipulator end-effector are quickly converged to zero. It proves the designed controller is efficient for the trajectory tracking of the non-zero velocity mobile manipulator's end-effector

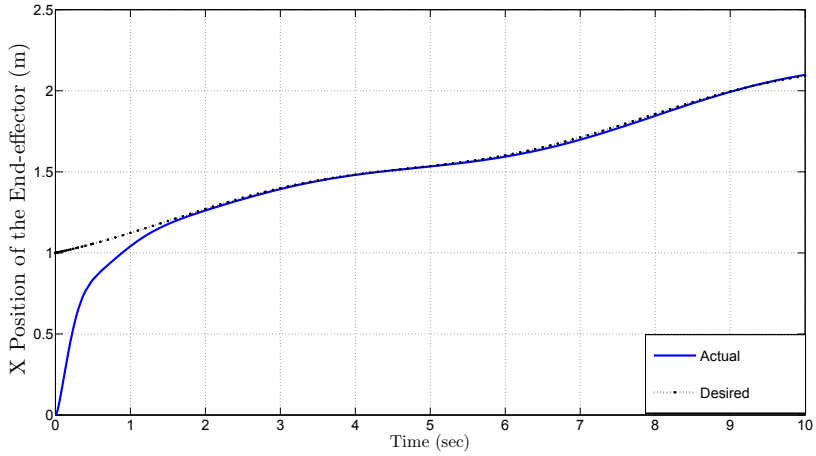


Figure 5.8: Desired and actual $x_{EE}$ for Case 1

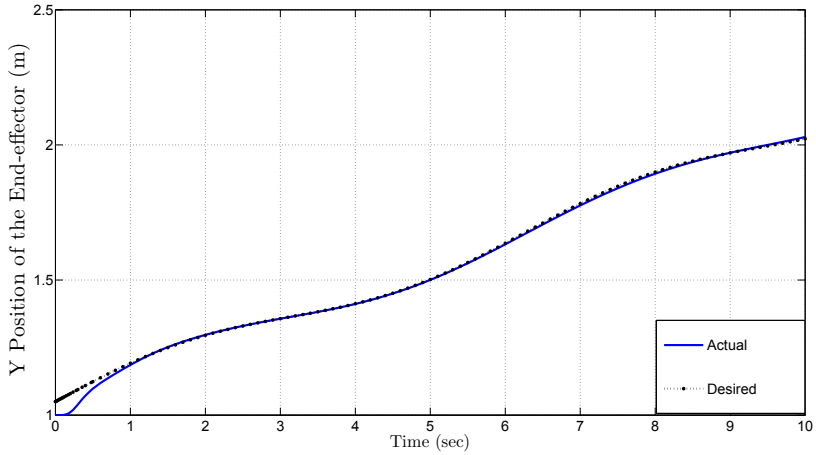


Figure 5.9: Desired and actual $y_{EE}$ for Case 1

position control. The desired and actual position of the end-effector in three dimensions are shown in Figs.5.12 and 5.13, where Fig.5.12 shows the desired and actual position in the XY plane. The mobile manipulator's end-effector is initially placed on $\left(x_{EE}, y_{EE}\right) = (0, 1)$. It follows the desired trajectory smoothly with a minimum error. Fig.5.13 shows the three dimensional view of the desired and actual position trajectory. Some data points are shown to compare both trajectories. Initially the end-effector of the mobile manipulator is at $\left[x_{EE}, y_{EE}, z_{EE}\right] = [0, 1, 2]$, then it tracks the desired trajectory and after 10 seconds the actual position of the end-effector is at $\left[x_{EE}, y_{EE}, z_{EE}\right] = [2.091, 2.022, 1]$ while the desired position is at

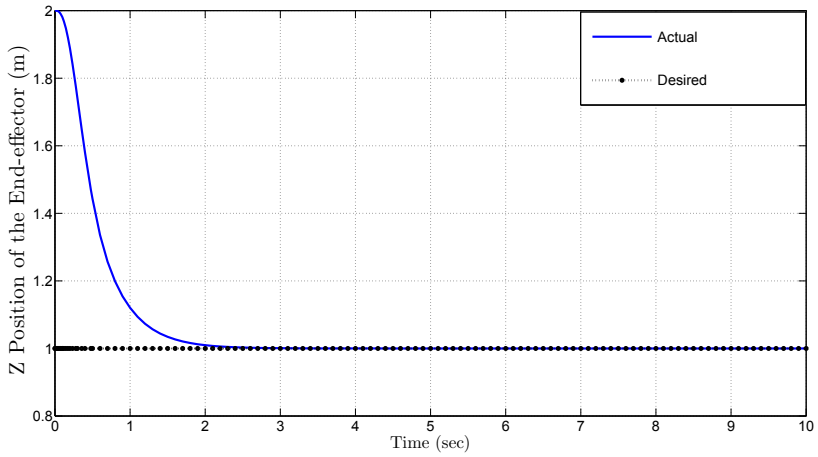


Figure 5.10: Desired and actual $z_{EE}$ for Case 1

$\left[x_{EE_d}, y_{EE_d}, z_{EE_d}\right] = [2.092, 2.023, 1]$ which proves the desired trajectory is almost achieved and the tracking error is very low and acceptable. Desired and actual velocities of the mobile manipulator in x and y directions are shown in the Figs.5.14 and 5.15. These velocities also converge to the desired velocities very quickly. After 1.5 seconds, mobile base velocity in x direction converge to the desired velocity. This small error is because of the redundancy resolution method used. This method cannot guarantee ideal solution to the inverse kinematics, it provides the approximation to the inverse kinematics and it helps to remove the singularity from the system. This kind of errors can be lowered by using proper redundancy resolution methods. However it cannot guarantee for the error-free trajectories because it deals with the singularities and redundancy.

Fig. 5.16 shows the actual joint velocities of the the mobile base and the first and second joint of the manipulator arm. Fig. 5.17 gives the plot of the controlled joint velocities of the mobile base and both joints. From both figures it can be seen that the controlled velocities $\eta_c$ are the same as that of the actual velocities $\eta$ of the joint. The ultimate goal is to control the joint velocities and it can be clearly identified from both figures that both velocities are same. The error between these velocities are plotted in Fig. 5.18. The error is zero indicates that both the actual and controlled joint velocities are same. Conclusions can be made from these results about the effectiveness of the backstepping kinematic controller. $\eta_c$ is the controlled joint

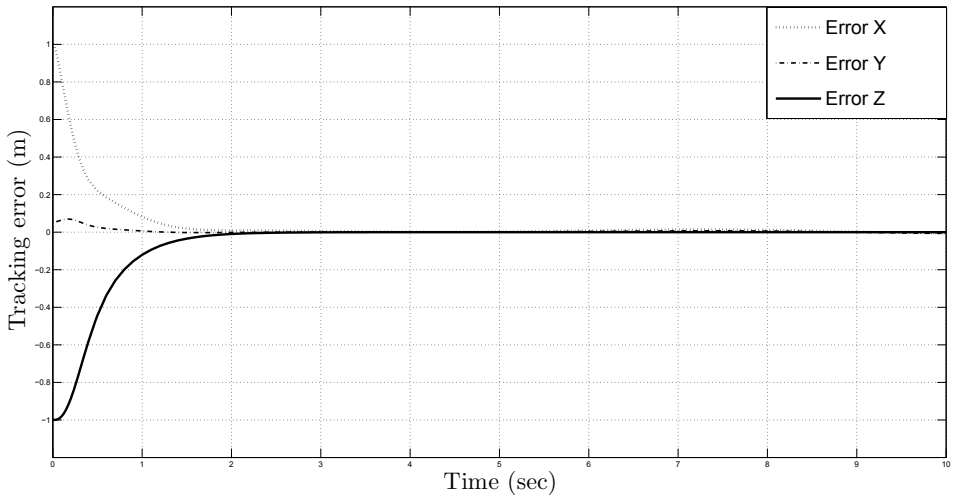


Figure 5.11: Tracking error of end-effector position for Case 1

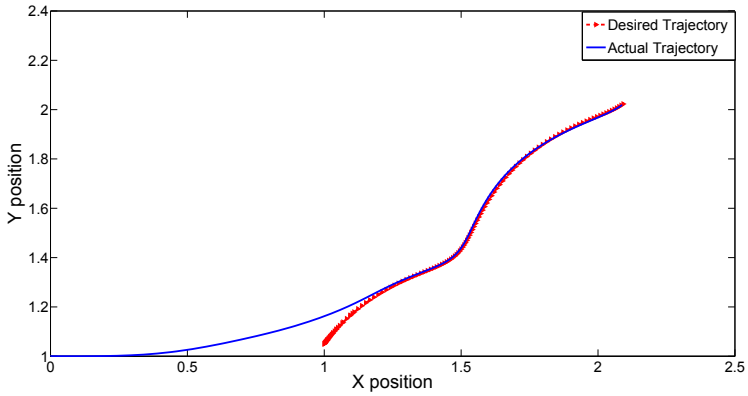


Figure 5.12: Desired and actual end-effector trajectory in XY plane for Case 1

velocities and it is derived using the backstepping velocity controller. $\eta$ is the actual joint velocity of the dynamic system of the mobile manipulator. Fig. 5.19 shows the estimated parameters of the system. These parameters are not the actual values of the respective parameters but they help achieve the zero velocity errors. The adaptive method chosen here is direct adaptive and in this method, the estimated parameter doesn't converge to their true value instead it helps the controller to provide zero error tracking. Fig.5.20 shows the input torque to the actuators. Fig.5.21 shows the torques over the 5 seconds of time. It is more clear from the figure that the the input torques quickly converges to constant values and bounded to that constant value.

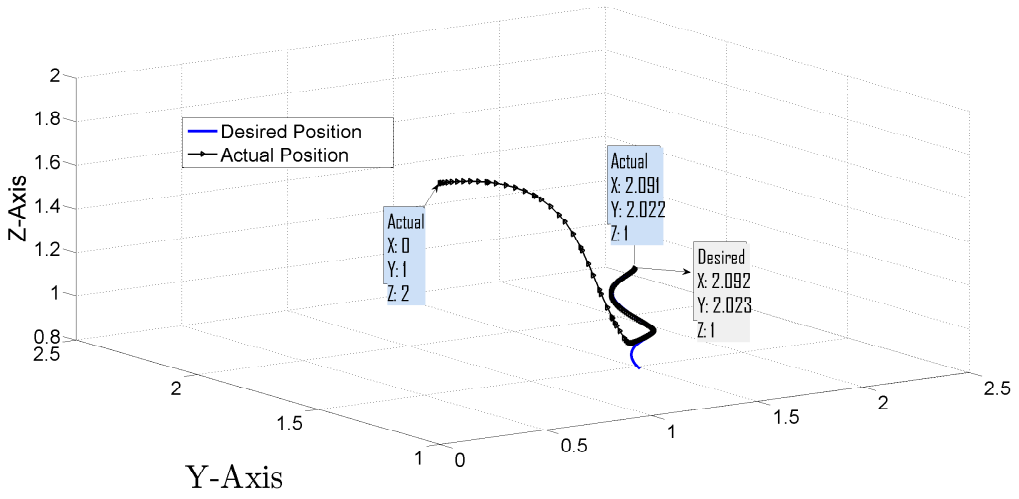


Figure 5.13: Desired and actual end-effector trajectory in the 3D plane for Case 1

### 5.2.2 Trajectory Tracking Case 2

Considering the other time dependent desired trajectory $\Psi_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}, z_{EE_d}\right]^T$ as Eq.(5.2).

$$\begin{cases} \dot{x} & = 2\omega\cos(\omega t) \\ \dot{y} & = 2\omega\sin(\omega t) \\ x_{EE_d} & = \sin(\omega t) \\ y_{EE_d} & = 1-\cos(\omega t) \\ z_{EE_d} & = 1 \end{cases} \tag{5.2}$$

$\omega = 0.1\pi$ is selected for this simulation. Note that the end-effector trajectory in $XY$ plane is a circular trajectory. The initial position and control gain used in this simulation procedure are given in Table 5.5 and 5.6. The simulation run-time for this simulation is chosen as 80 seconds. The parameters and the system model used is exactly the same as the section 5.2.1.

$X$, $Y$ and $Z$ components of the desired trajectory of the end-effector can be shown in Fig.5.22, where the desired $X$ and $Y$ components are sinusoidal and desired $Z$ component is constant.

The actual and desired trajectories of the end-effector in $X$, $Y$ and $Z$ directions are shown in Figs.5.23, 5.24 and 5.25 respectively. In Fig.5.23, the actual $X$ position of

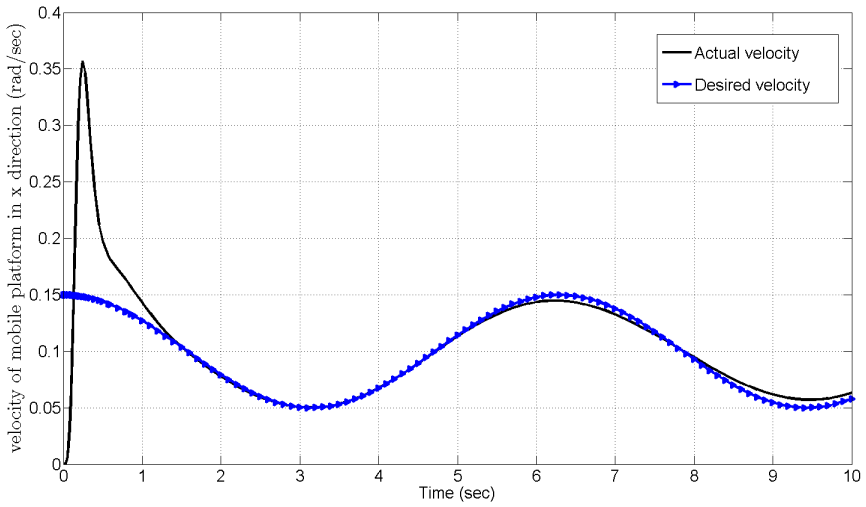


Figure 5.14: Desired and actual mobile base velocity in x direction for Case 1

Table 5.5: Initial position of the end-effector of the mobile manipulator for Case 2

| State | Initial Position |
|---|---|
| $\dot{x}$ | 0 |
| $\dot{y}$ | 0 |
| $x_{EE_d}$ | 0 |
| $y_{EE_d}$ | 1 |
| $z_{EE_d}$ | 2 |

Table 5.6: Control gain values for Case 2

| Control Gain | Value |
|---|---|
| $k$ | 1.5 |
| $K$ | 0.2 |
| $K_1$ | $250I_{4\times4}$ |
| $\Gamma$ | $-10I_{6\times6}$ |

the end-effector tracks the desired $X$ position in 20 seconds and then converge to the desired $X$ position throughout. Fig.5.24 compares desired and actual $Y$ position of the end-effector. The actual $Y$ response starts tracking the desired $Y$ position within 25 seconds of time and after that it converges to the desired position. However, there is a steady error throughout because of the redundancy resolution method used. Fig.5.25 represents the actual and desired $Z$ position of the end effector. From the figure, it almost 40 seconds for it to converge to the desired constant value. It is shown in

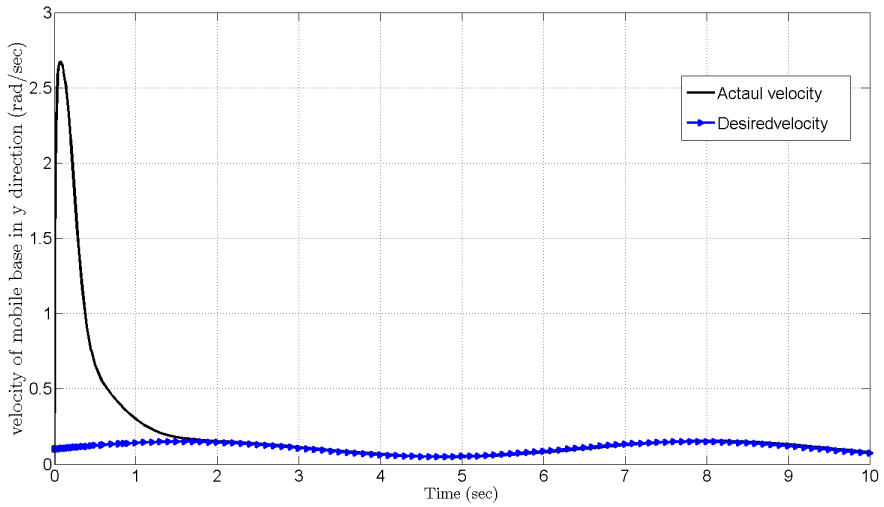


Figure 5.15: Desired and actual mobile base velocity in y direction for Case 1

Figs.5.23, 5.24 and 5.25 that the $Z$ position takes more time than $X$ and $Y$ positions to converge to the desired position.

The trajectory tracking error of the Case 2 is given in Fig.5.26. All error signals converge to zero but the $Y$ signal is not converging to exact zero value; and there is 5 percentage error at 50 seconds which is almost 2 percentage at 70 seconds with respect to the desired values. As discussed earlier, the DLS method cannot provide ideal solution to the inverse kinematics of the mobile manipulator system. This error can be reduced by increasing the simulation run time. However there is a 2 percentage of error after running a simulation for long time and this error becomes constant. However for some applications, where accuracy is the main concern, for example, surgery robots, this error cannot be ignored and this can be the limitation of using the DLS method for the redundancy resolution. $X$ and $Y$ position errors of the mobile manipulator end-effector are converging to zero.

The trajectory tracking in XY and XZ plane are given in the following figures. Fig.5.27 shows the desired and actual trajectory of the end-effector in $XY$ plane. The actual end-effector position starts from $(0, 1)$ and tracks the desired circular trajectory as shown in the figure. The desired trajectory is followed by the actual trajectory well. A clearer view of the figure is plotted in Fig.5.28. Fig.5.29 shows the desired and actual position of the end-effector in XZ plane. Initially the actual $Z$ position

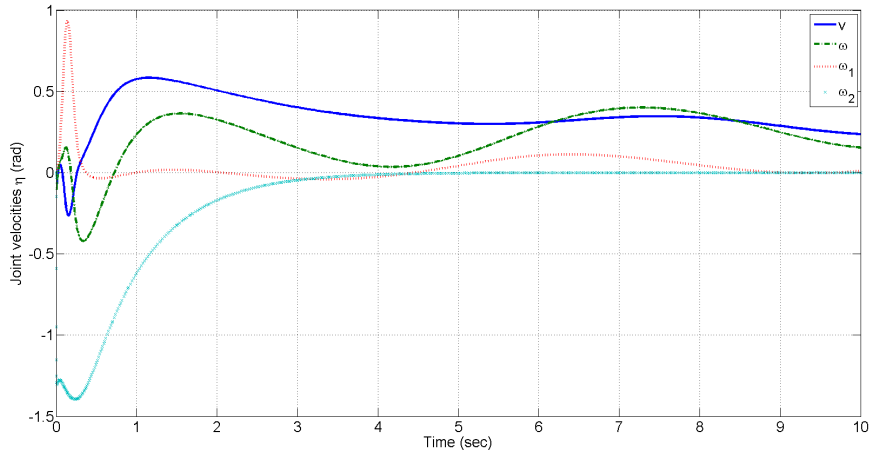


Figure 5.16: Actual velocities of the operational space joints for Case 1

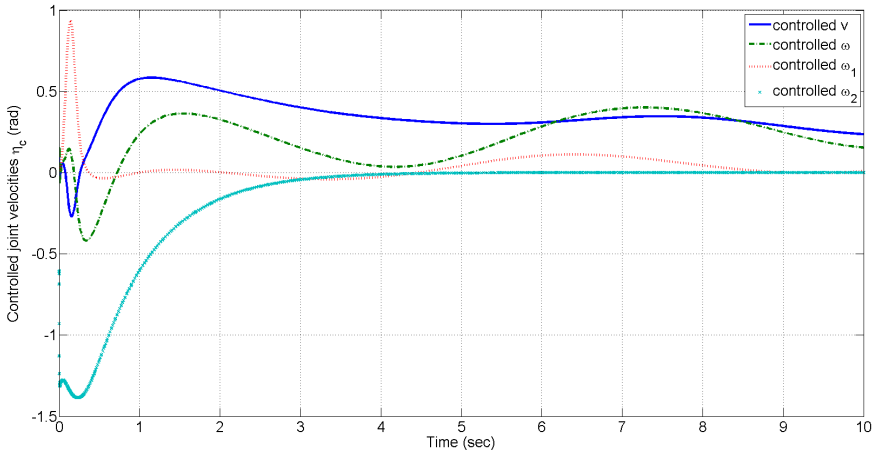


Figure 5.17: Controlled velocities of the operational space joints for Case 1

of the end-effector is at 2 and then it tracks the desired trajectory as shown in the figure.

The actual and desired velocity of the mobile base in X and Y direction are given in Figs.5.30 and 5.31. Fig.5.30 shows the desired and actual mobile base velocity in $X$ direction.The actual velocity converges to the desired velocity in 25 seconds and it smoothly converges to the desired velocity in $X$ direction. Similarly an actual mobile base velocity tracks the desired velocity in $Y$ direction in 25 seconds smoothly. These figures shows how the mobile base velocity converges to the desired velocity of the mobile base.

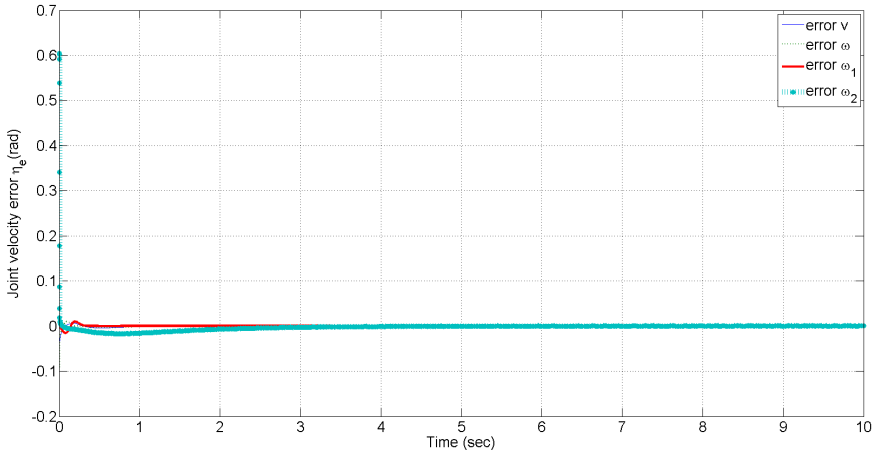


Figure 5.18: Error velocities of the operational space joints for Case 1

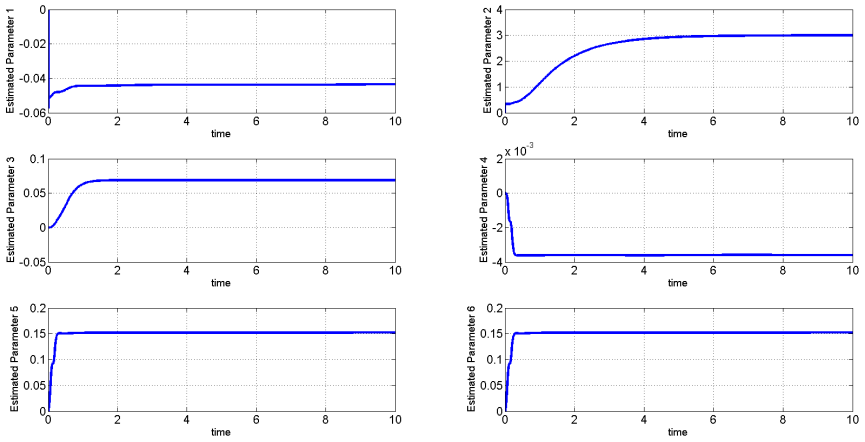


Figure 5.19: Estimated parameters for Case 1

Figs.5.32 and 5.33 show the actual velocity $\eta$ and controlled velocity $\eta_c$ of the mobile manipulator joints. From the figures, it is clear that both the actual and controlled velocities are same and there is very low error between them. The right and left wheel velocities are same in this case because of the circular trajectory. Fig.5.34 gives the error of the actual and controlled joint velocities of the mobile manipulator. The error is very small about 0.0001 rad/sec for the right and left wheel velocities. It is 0.000025 rad/sec for the first link joint velocity and almost 0 for the second link joint velocity. These zero errors mean that both the controlled and actual joint velocities of the mobile manipulator are same.

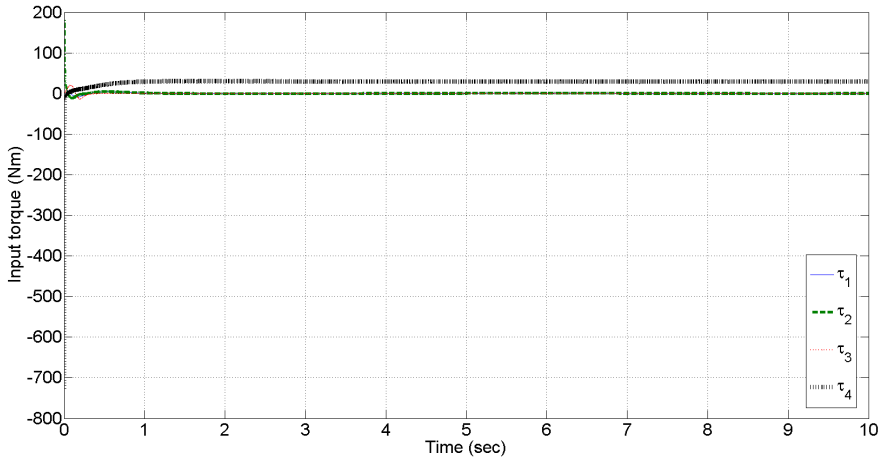


Figure 5.20: Input torque for Case 1

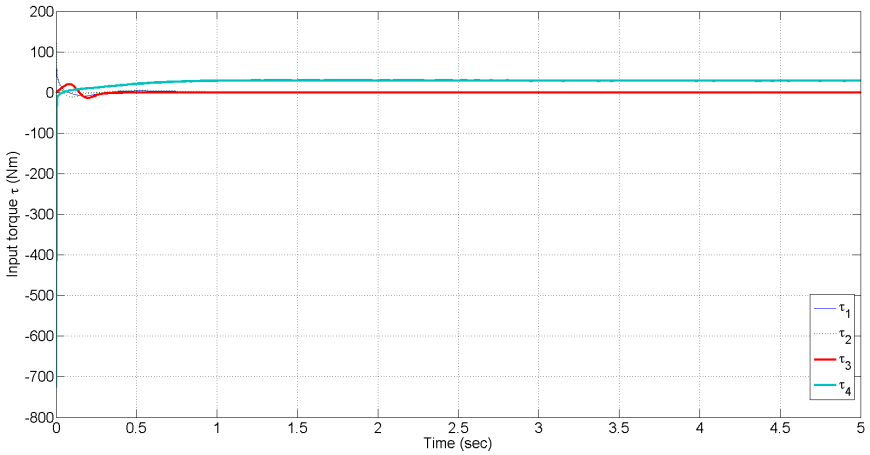


Figure 5.21: Input torque (5 sec) for Case 1

Fig.5.35 illustrates the estimated parameters of the system over the time. As seen in the figure, the parameter converges to a constant value $\mathbf{p} = [1.4 \times 10^{-5}, 1, 5.9 \times 10^{-3}, -1.4 \times 10^{-5} 9.5 \times 10^{-3}, 9.45 \times 10^{-3}]^T$.This adaption of the parameters to some constant value is absolutely what is expected of the control scheme used here. The settling time is quick (smaller than 40 seconds) and and parameter value is constant. As discussed in Section 5.2.1, it is significant to indicate that these parameters do not actually represent the true value of the parameters. These parameters are adapted to ensure the limited tracking errors only. To get the true convergence of the parameter,

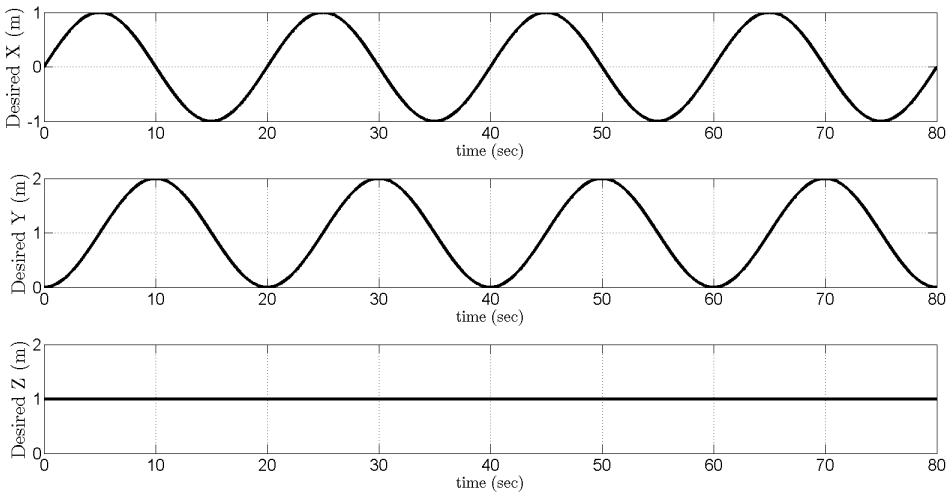


Figure 5.22: Desired end-effector position trajectories for Case 2

an indirect adaptive method should be chosen, which would adapt the true parameter values. However this method (direct adaptive) maintains the bounded tracking errors. The control law defined here requires the complete knowledge of the mobile manipulator system, i.e. dynamics, kinematics and inverse kinematics. If these are not completely and accurately known, the performance of the control law will be ineffective over time. The lower performance of the controller should also be expected for the external disturbance case because of the incomplete knowledge of the mobile manipulator system.

Fig.5.36 shows the input torque of the joints $\bar{\tau}$.

### 5.2.3 Trajectory Tracking Case 3

In previous sections, mobile manipulator's output operational space has five states and the DLS method is used to remove the singularity from the system. However, in this case, we assume the operational space has only four states and therefore the Jacobian can be inverted using the matrix laws. However the system is still singular. The inverse of Jacobian is not possible because the $4 \times 4$ Jacobian matrix is full rank and therefore it is subjected to singularities. Similar to the first two cases, a redundancy resolution method is used to invert the Jacobian of the mobile manipulator system. In the modeling process, it is assumed that the manipulator arm is mounted at the

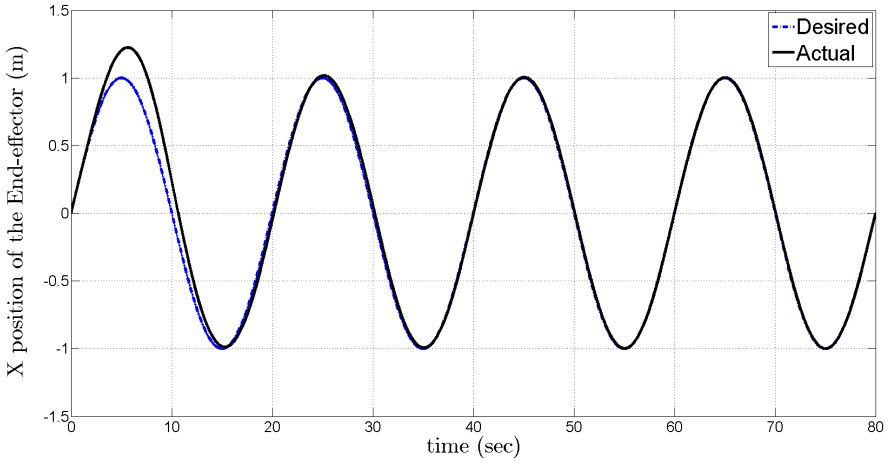


Figure 5.23: Desired and actual $x_{EE}$ for Case 2

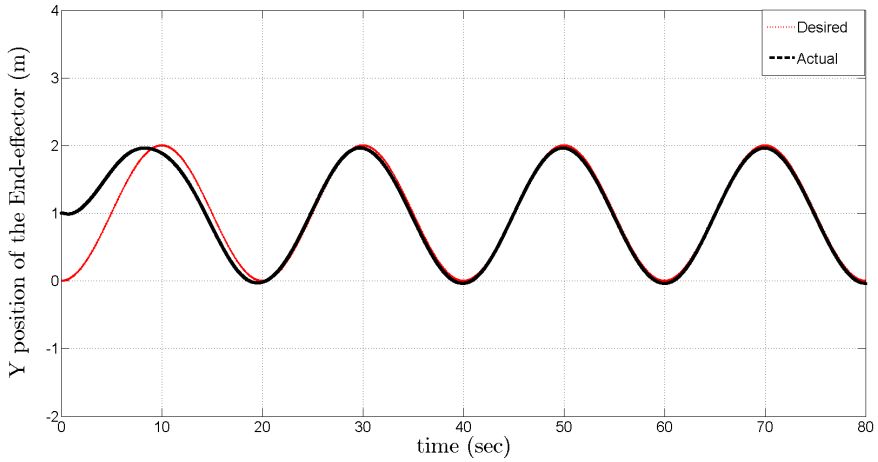


Figure 5.24: Desired and actual $y_{EE}$ for Case 2

center of the wheel axis and because of that two columns of Jacobian matrix are same and this makes the system subjected to singularities. Four states are considered for the trajectory tracking of Case 3.The desired trajectory $\Psi_d = \left[\dot{x_d}, \dot{y_d}, x_{EE_d}, y_{EE_d}\right]^T$ is selected as in Eq.(5.1) ignoring the fifth variable $Z$. Also, instead the left wheel and right wheel velocity of the mobile base, the linear and angular velocities of the mobile base are used for the velocity commands. The control gain values chosen are given in Table 5.7.

The initial position of the mobile manipulator system remains same as Table 5.3 ignoring position of $z_{EE}$. The actual and desired $x_{EE}$ position and $y_{EE}$ positions are

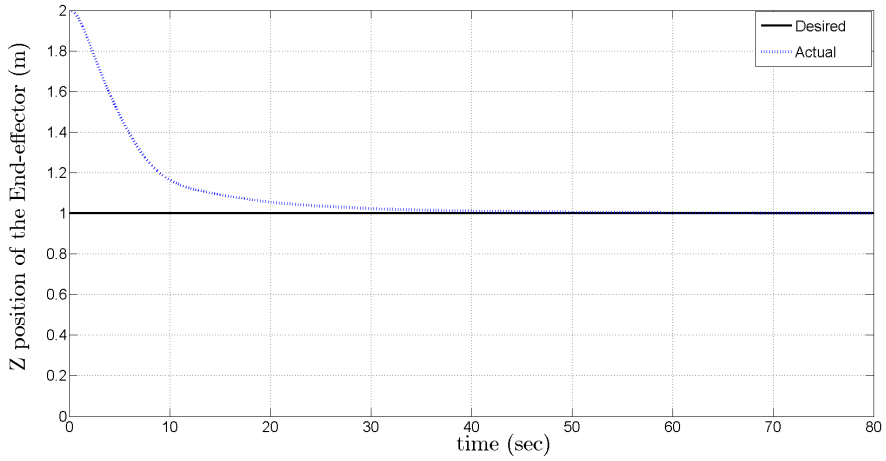


Figure 5.25: Desired and actual $z_{EE}$ for Case 2

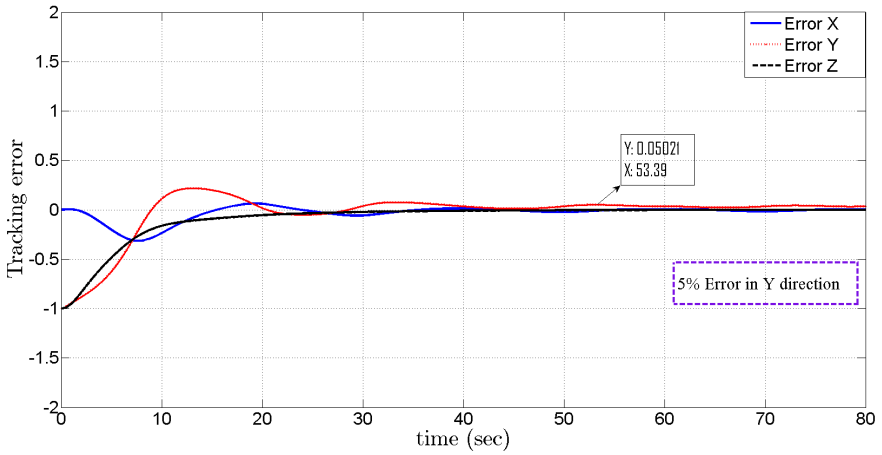


Figure 5.26: Tracking error of end-Effector position for Case 2

Table 5.7: Control gain values for Case 3

| Control Gain | Value |
|---|---|
| $k$ | 0.05 |
| $K$ | 1.2 |
| $K_1$ | $80I_{4\times4}$ |
| $\Gamma$ | $-10I_{8\times8}$ |

given in Figs.5.37 and 5.38. The actual responses track the desired responses smoothly and quickly. However, it takes more time to converge to the desired value than Case 1 but it converge to the desired trajectory smoothly and accurately with less error comparatively. This is because the controller has less states to track and therefore

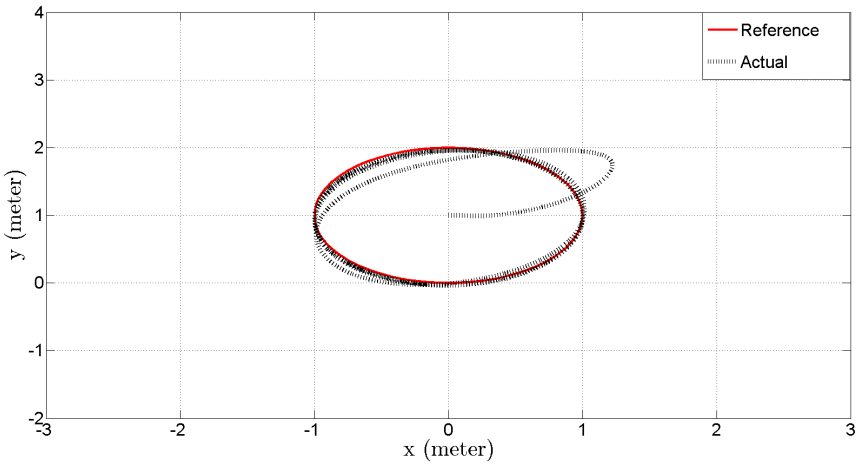


Figure 5.27: Actual and desired trajectory tracking of the Case 2 in XY plane

the error is comparatively very small. The value of $k$ is smaller which ensures the results near the singular configuration.

The error between the desired and actual positions are given in Fig.5.39. The errors converges to zero smoothly.

The Actual and desired velocity of mobile base in $x$ and $y$ directions are shown in Figs.5.40 and 5.41. Both velocities converges to the desired trajectory. Comparing it with Case 1, it has very small peak values of the velocities. This confirms that the damping factor $k$ values are near the singular configuration.

The actual joint velocity of the mobile manipulator is plotted in Fig.5.42. The controlled joint velocities $\eta_c$ is plotted to compare the controlled velocity to actual joint velocities. From Fig.5.44, the controlled velocities track the actual velocities and the error is almost zero.

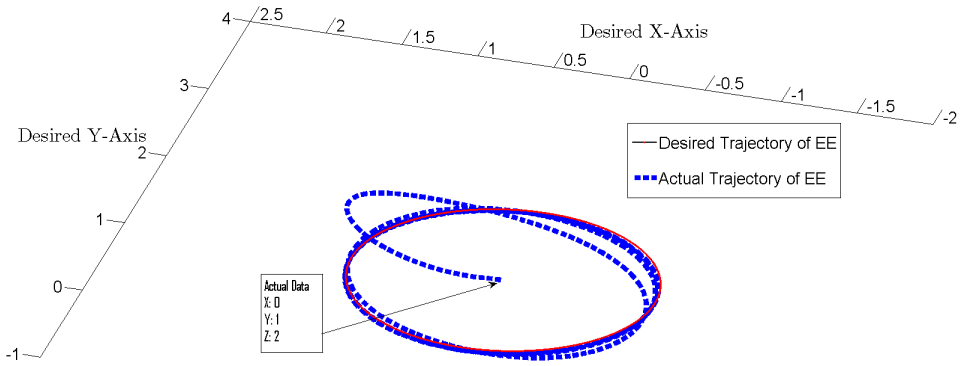


Figure 5.28: Actual and desired trajectory tracking of the Case 2 in XY plane

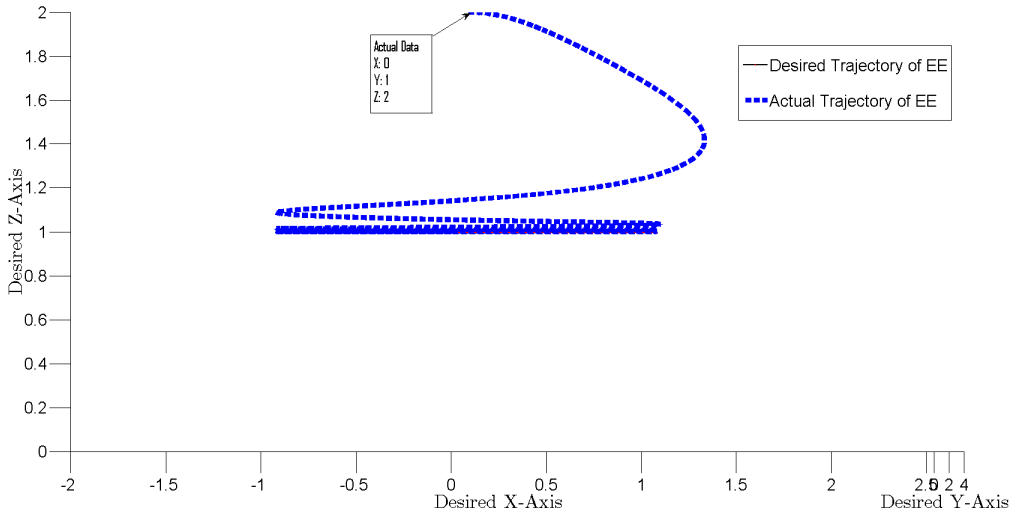


Figure 5.29: Actual and desired trajectory tracking of the Case 2 in XZ plane

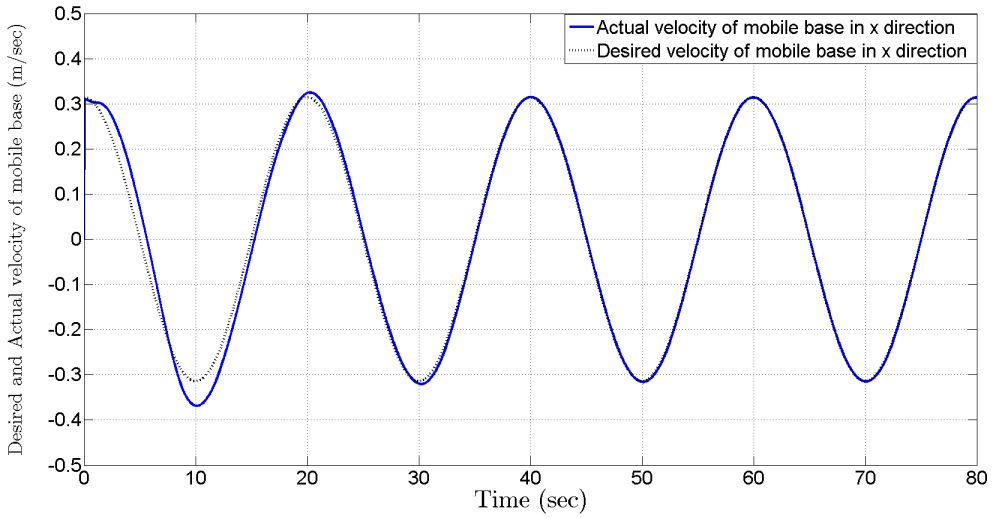


Figure 5.30: Desired and actual $\dot{x}$ for Case 2

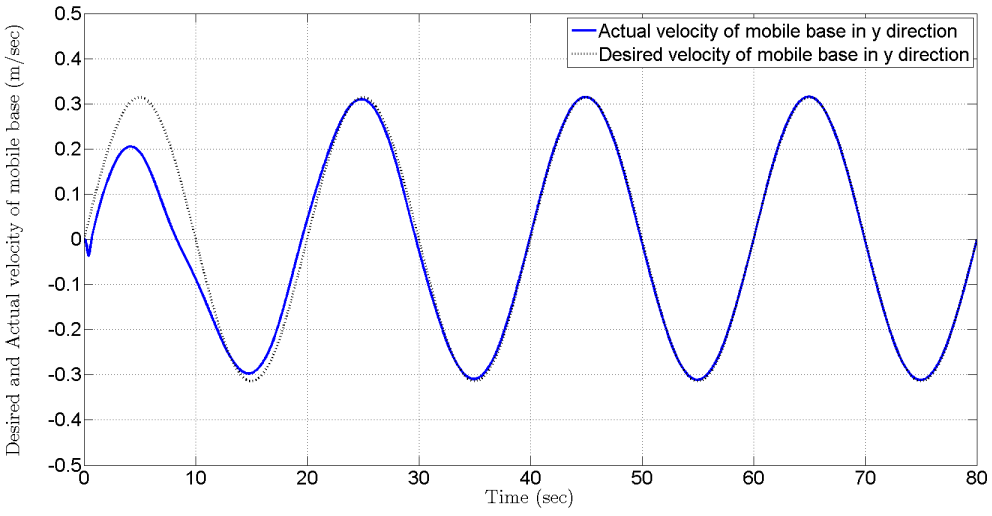


Figure 5.31: Desired and actual $\dot{y}$ for Case 2

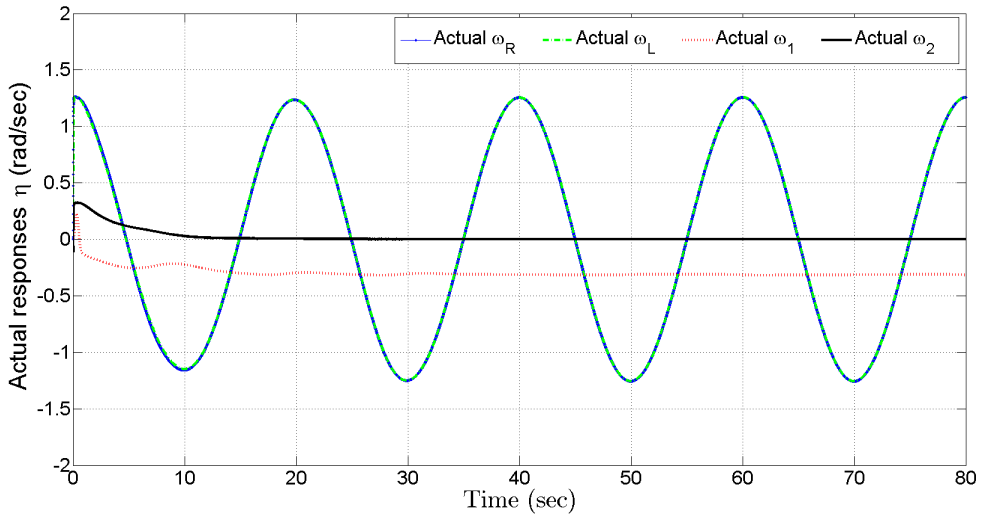


Figure 5.32: Actual velocities of the operational space joints for Case 2

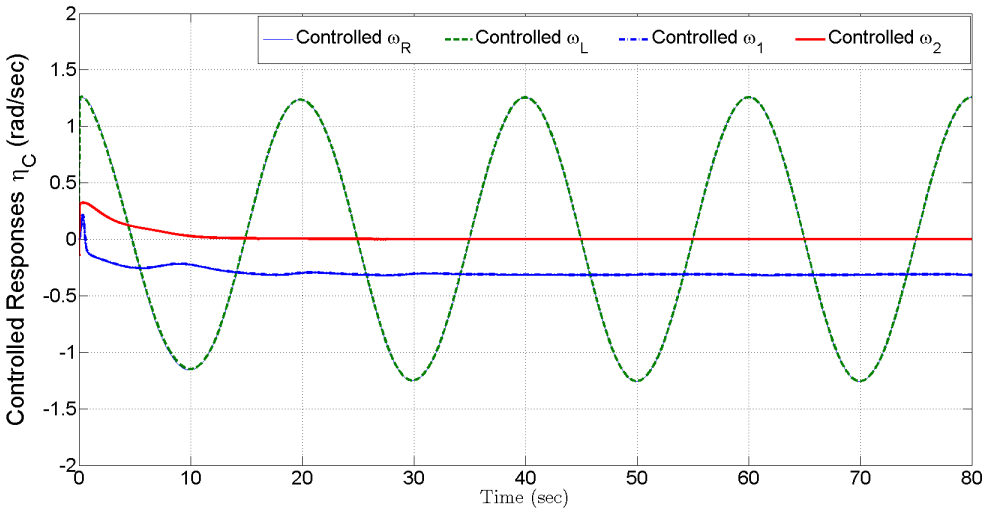


Figure 5.33: Controlled velocities of the operational space joints for Case 2

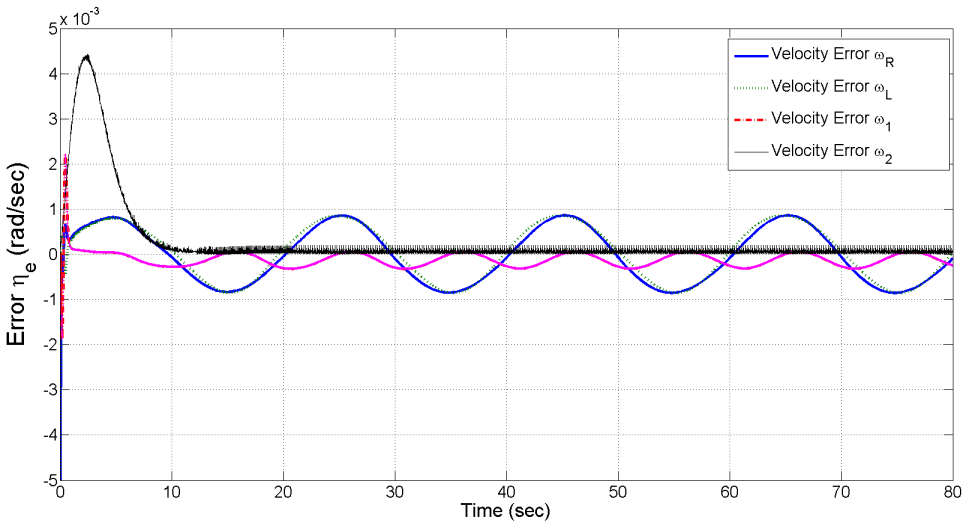


Figure 5.34: Error velocities of the operational space joints for Case 2

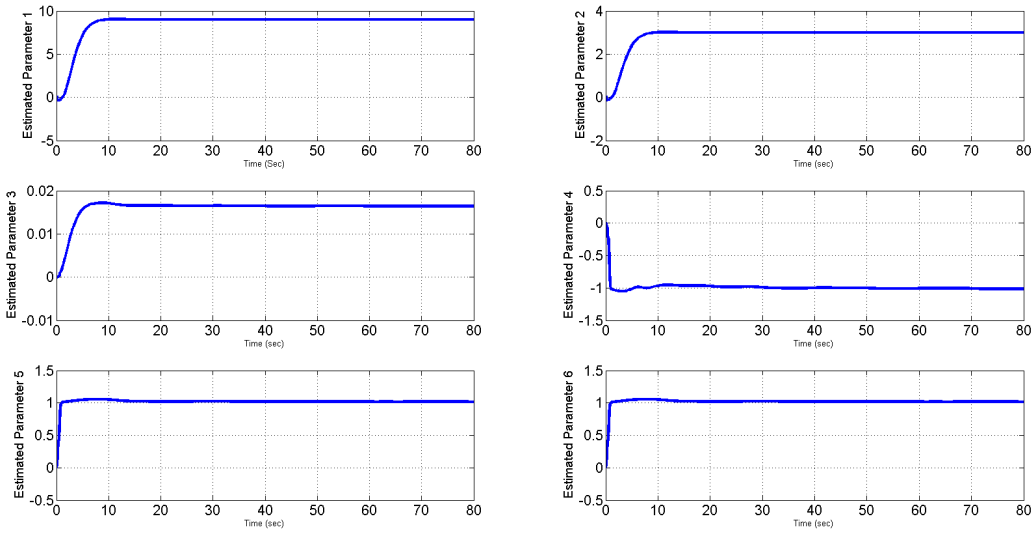


Figure 5.35: Estimated parameters for Case 2

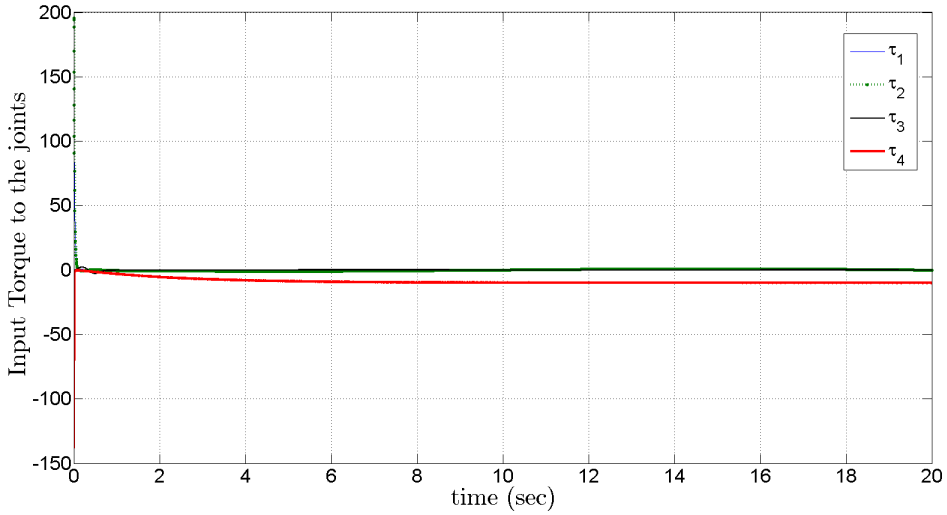


Figure 5.36: Input torque of the joints for Case 2

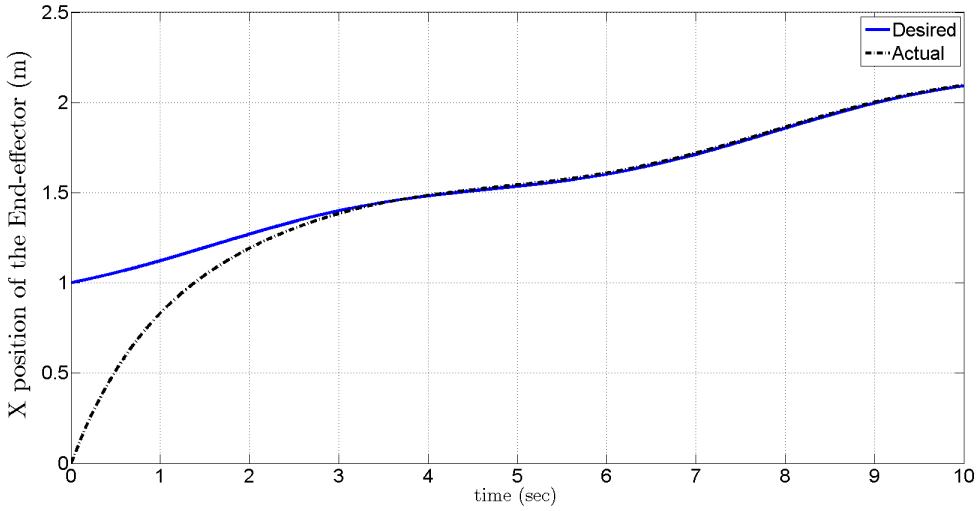


Figure 5.37: Actual and desired x positions for Case 3

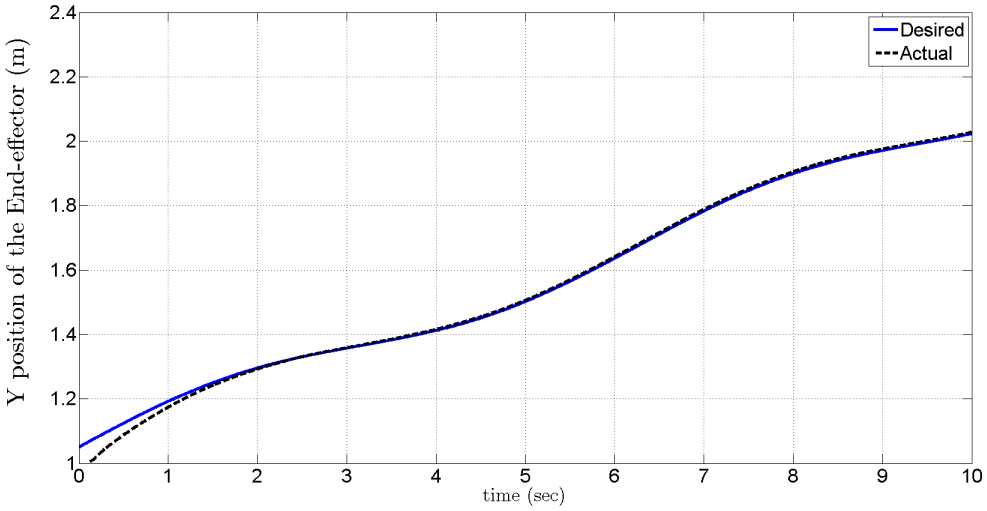


Figure 5.38: Actual and desired y positions for Case 3

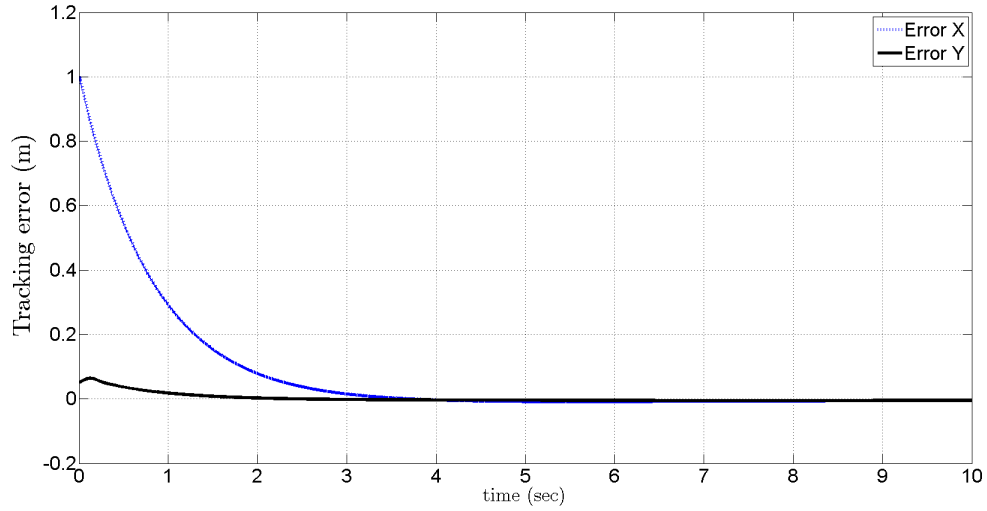


Figure 5.39: Tracking errors for Case 3

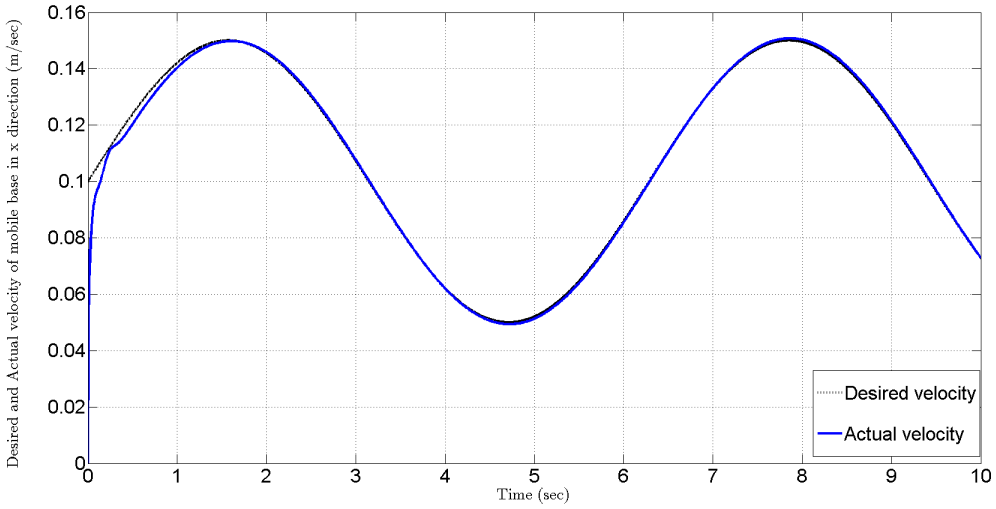


Figure 5.40: Actual and desired velocities of mobile base in x direction for Case 3

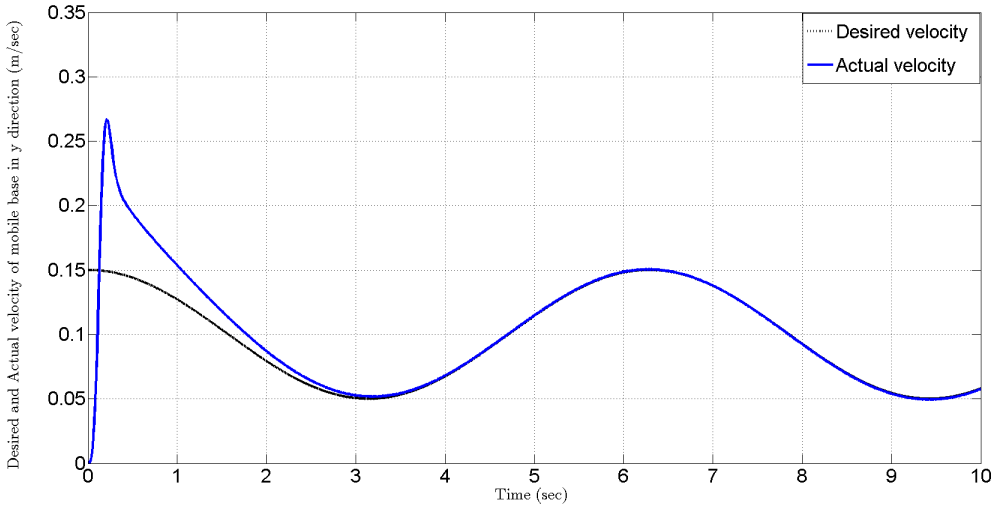


Figure 5.41: Actual and desired velocities of mobile base in y direction for Case 3

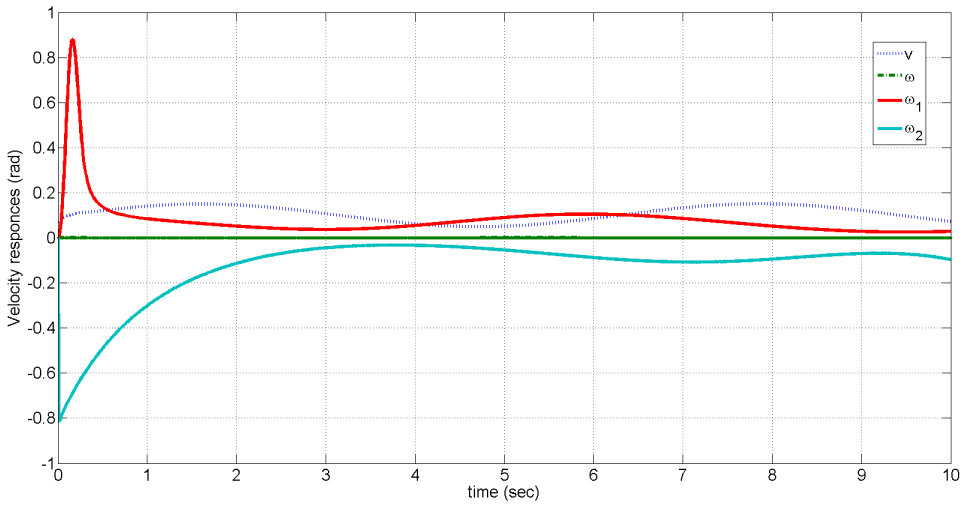


Figure 5.42: Actual joint velocities for Case 3

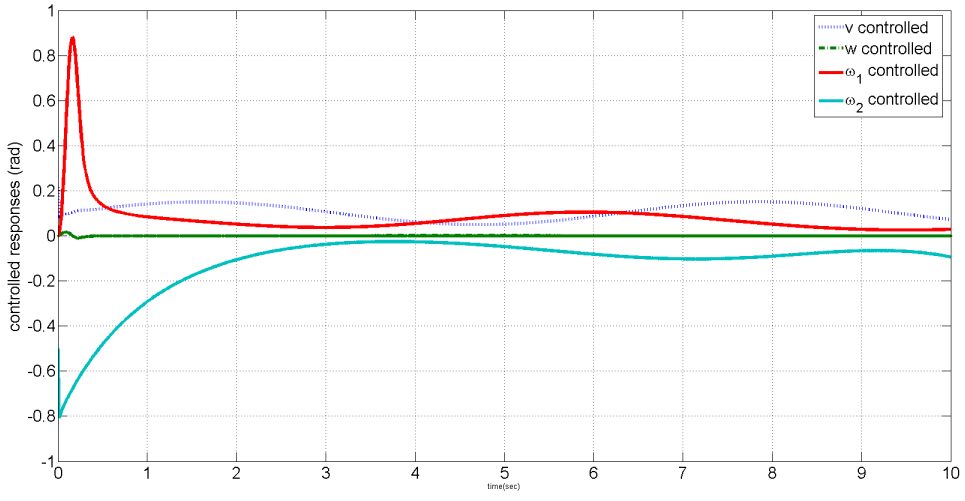


Figure 5.43: Controlled joint velocities for Case 3

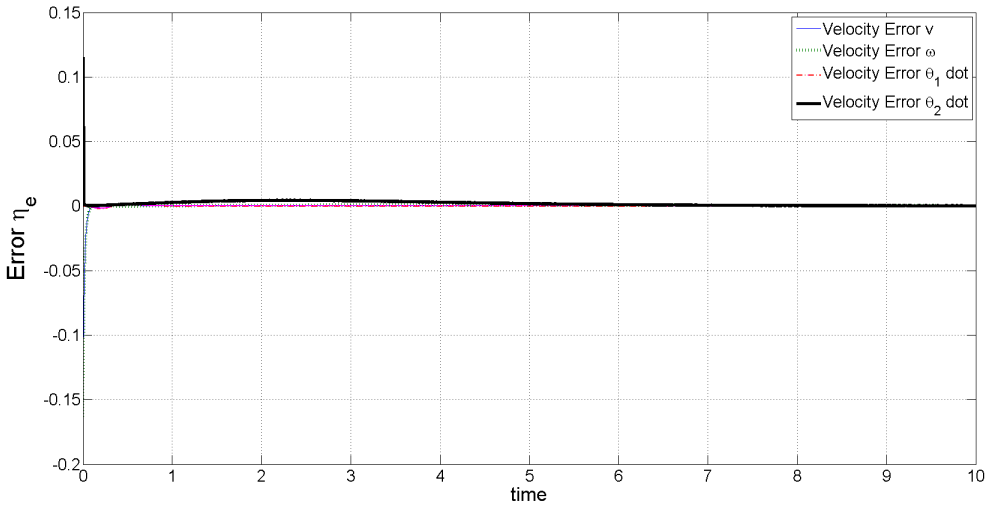


Figure 5.44: Error of joint velocities for Case 3

The estimated parameter values converge to a constant value $\mathbf{p} = [-0.135, -0.135, -0.135, 0.39, 0.9, -3.7 \times 10^{-3}, 0.24, 0.86]^T$ shown in Fig.5.45. The input torque of the joint for Case 3 is shown in Fig.5.46. The large joint torques in the beginning converge to a value of 9.434 and bound.

Simulation studies have verified that the system states converge to the desired position and velocity trajectories with minimum errors. Redundancy and singularity of the mobile manipulator system are resolved by DLS method which provides satisfying results. The input torque is bounded and converge to a constant value for the backstepping-adaptive controller whereas in MBC, the torque is bounded but fluctuating for large values. Three cases are considered for the simulation of a 5-DOF mobile manipulator subjected to nonholonomic constraints, redundancy and singularity. Simulation studies have verified the effectiveness of the backstepping-adaptive controller.

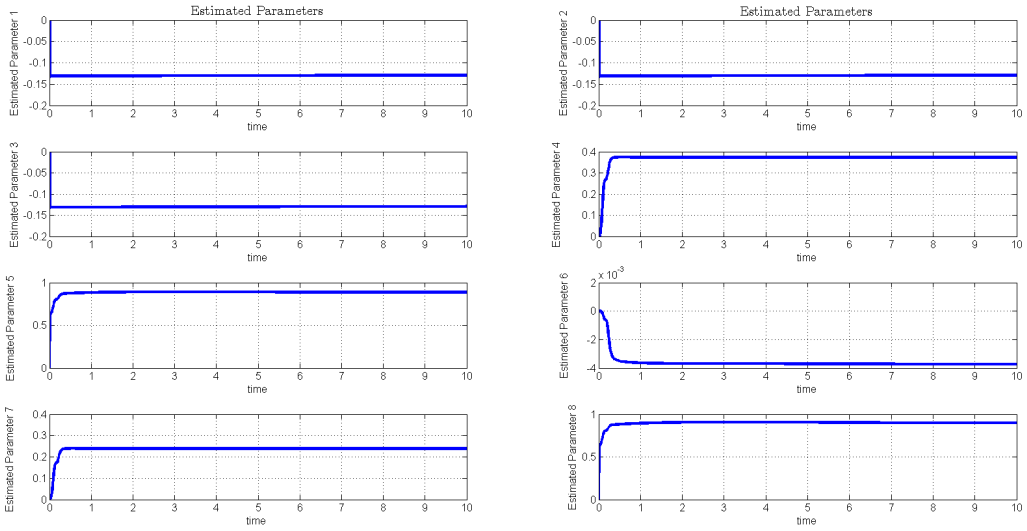


Figure 5.45: Estimated paramters for Case 3

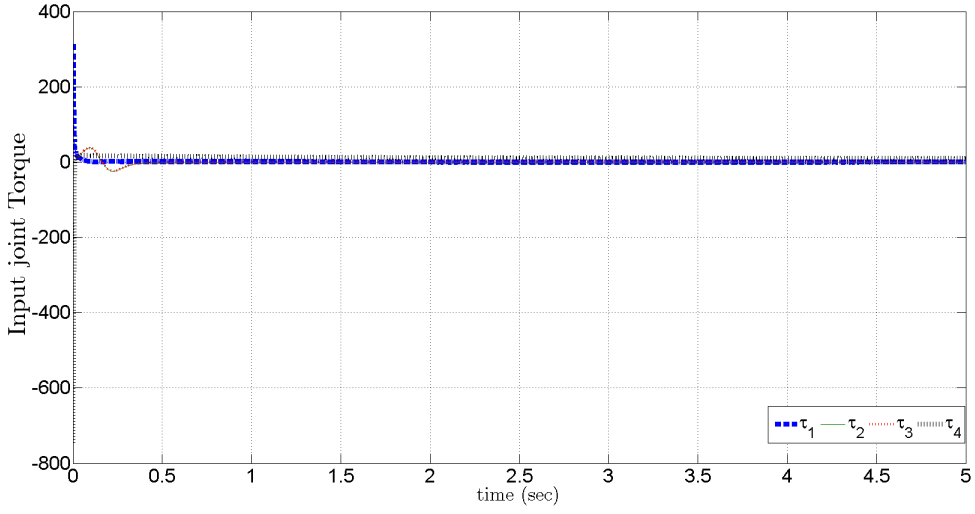


Figure 5.46: Input torques for each joints for Case 3

# Chapter 6

# Conclusions and Future Work

## 6.1 Conclusions

The goal of this dissertation was to design a systematical mathematical modeling of the mobile manipulator system and to design a backstepping adaptive controller efficient to provide an n-link mobile manipulator with an objective of highly precise simultaneous trajectory tracking of mobile base velocity and the position of the end-effector. This research work was attempted as a result of the high level of interest in the area of control of mobile manipulators. Several modeling and control techniques were examined before, however the main focus was expeditiously narrowed down to the adaptive control algorithm. The purpose of this research was to develop a novel approach by improving a standard adaptive control algorithm. We have determined a new control design strategy. This controller have been found to be beneficial to the control problem of mobile manipulator with unknown parameters. A kinematic backstepping controller was designed followed by an adaptive torque controller so that a mobile manipulator trajectories converge to the desired trajectories. Redundancy resolution method was also used to resolve the redundancy of the mobile manipulator.

Following the designed control scheme and redundancy resolution method an extensive simulation study was performed in order to ensure the efficiency of the designed controller. A 5-DOF nonholonomic mobile manipulator was used in the simulation study. The complete kinematics and dynamic model was derived in order to perform the simulation. The results of the simulation study have indeed proved the effectiveness of the proposed controller. The following conclusions can be made from the simulation studies:

The proposed controller achieves the very good trajectory tracking performance without generating unnecessary high frequency torques comparing to the other controllers in literature and case studies.

## 6.2 Future Work

Although the overall goal of the dissertation was achieved, there are still many improvements that could be made in order to make this controller a realistic controller for real application. Several of these improvements would deal with the ability of the controller to respond reliably to external disturbances and unknown model dynamics, giving the controller wider range and reliability. By applying the suitable redundancy resolution method along with the proposed controller, a better performance can be achieved. The new robust controller will be designed and will be tested on the experimental mobile manipulator. A case of cooperative control can be further developed for multiple mobile manipulators.

## Bibliography


[1] S. Cubero, ed., *Industrial Robotics: Theory, Modelling and Control*. Pro Literatur Verlag, Germany / ARS, Austria, dec 2006.

[2] The National Aeronautics and Space Administration (NASA), "GPN-2000-001817 - Puma Robotic Arm," 1990. [Online; accessed 26-July-2014].

[3] Wikipedia, "Classic-dhparameters," 2014. [Online; accessed 21-July-2014].

[4] Z. Li, S. S. Ge, and A. Ming, "Adaptive robust motion/force control of holonomic-constrained nonholonomic mobile manipulators," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 3, pp. 607–616, 2007.

[5] S. Zhou, Y. C. Pradeep, and P. C. Chen, "Simultaneous base and end-effector motion control of a nonholonomic mobile manipulator," in *Automation, Robotics and Applications (ICARA), 2015 6th International Conference on*, pp. 143–148, IEEE, 2015.

[6] J. Jimenez-Lozano and B. Goodwine, "Nonlinear disturbance decoupling for a mobile robotic manipulator over uneven terrain," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6930–6936, 2011.

[7] K. Tchon and R. Muszynski, "Instantaneous kinematics and dexterity of mobile manipulators," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 3, pp. 2493–2498, IEEE, 2000.

[8] K. Tchoń and J. Jakubiak, "Endogenous configuration space approach to mobile manipulators: a derivation and performance assessment of jacobian inverse kinematics algorithms," *International Journal of Control*, vol. 76, no. 14, pp. 1387–1419, 2003.

[9] B. D'andréa-novel, G. Campion, and G. Bastin, "Control of wheeled mobile robots not satisfying ideal velocity constraints: a singular perturbation approach," *International Journal of Robust and Nonlinear Control*, vol. 5, no. 4, pp. 243–267, 1995.

[10] A. De Luca, G. Oriolo, and P. R. Giordano, "Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 1867–1873, IEEE, 2006.

[11] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," *Nonholonomic Motion Planning*, p. 271342, 1993.

[12] J.-P. Laumond, S. Sekhavat, and F. Lamiraux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot motion planning and control*, pp. 1–53, Springer, 1998.

[13] Y. Nakamura, "Advanced robotics: Redundancy and optimization, addison-wesley," *Reading, MA*, 1991.

[14] K. Gochev, A. Safonova, and M. Likhachev, "Planning with adaptive dimensionality for mobile manipulation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2944–2951, IEEE, 2012.

[15] S. Murray, W. Floyd-Jones, Y. Qi, D. Sorin, and G. Konidaris, "Robot motion planning on a chip," *Robotics: Science and System*, 2016.

[16] V. Andaluz, P. Leica, F. Roberti, M. Toibero, and R. Carelli, "Robust backstepping control of nonlinear systems using neural network," *Frontiers in Advanced Control Systems*, pp. 164–189, 2012.

[17] R. Morales, J. Somolinos, and J. A. Cerrada, "Dynamic control of a reconfigurable stair-climbing mobility system," *Robotica*, vol. 31, no. 2, pp. 295–310, 2013.

[18] R. W. Brockett *et al.*, "Asymptotic stability and feedback stabilization," *Differential geometric control theory*, vol. 27, no. 1, pp. 181–191, 1983.

[19] Y. Yamamoto and X. Yun, "Effect of the dynamic interaction on coordinated control of mobile manipulators," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 5, pp. 816–824, 1996.

[20] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot: backstepping kinematics into dynamics," in *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, vol. 4, pp. 3805–3810, IEEE, 1995.

[21] J. Tan, N. Xi, and Y. Wang, "Integrated task planning and control for mobile manipulators," *The International Journal of Robotics Research*, vol. 22, no. 5, pp. 337–354, 2003.

[22] M. Kristic and K. kanellakopoulos, "Nonlinear and adaptive control design," New York: Wiley, 1995.

[23] C. Kwan and F. L. Lewis, "Robust backstepping control of nonlinear systems using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 30, no. 6, pp. 753–766, 2000.

[24] R. Freeman and P. V. Kokotovic, *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science & Business Media, 2008.

[25] S.-C. Tong and Y.-M. Li, "Adaptive backstepping output feedback control for siso nonlinear system using fuzzy neural networks," *International Journal of Automation and Computing*, vol. 6, no. 2, pp. 145–153, 2009.

[26] M. Cheng and C. Tsai, "Robust backstepping tracking control using hybrid sliding-mode neural network for a nonholonomic mobile manipulator with dual arms," *IEEE CONFERENCE ON DECISION AND CONTROL*, vol. 44, no. 2, p. 1964, 2005.

[27] M. Boukottaya and T. Damak, "Robust adaptive control for mobile manipulators," *International Journal of Automation and Computing*, vol. 8, no. 1, pp. 8–13, 2011.

[28] G. White, *Simultaneous motion and interaction force control of a nonholonomic mobile manipulator*. PhD thesis, Citeseer, 2006.

[29] G. Feng and R. Lozano, *Adaptive control systems*. Newnes, 1999.

[30] Z. Li, S. S. Ge, M. Adams, and W. S. Wijesoma, "Robust adaptive control of uncertain force/motion constrained nonholonomic mobile manipulators," *Automatica*, vol. 44, no. 3, pp. 776–784, 2008.

[31] S. S. Ge, Z. Wang, and T. H. Lee, "Adaptive stabilization of uncertain nonholonomic systems by state and output feedback," *Automatica*, vol. 39, no. 8, pp. 1451–1460, 2003.

[32] M. Fang, W. Chen, and Z. Li, "Adaptive tracking control of coordinated nonholonomic mobile manipulators," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 4343–4348, 2008.

[33] H. G. Tanner, K. J. Kyriakopoulos, and N. Krikelis, "Modeling of multiple mobile manipulators handling a common deformable object," *Journal of Robotic Systems*, vol. 15, no. 11, pp. 599–623, 1998.

[34] A. Tayebi, "Adaptive iterative learning control for robot manipulators," *Automatica*, vol. 40, no. 7, pp. 1195–1203, 2004.

[35] C. Han, C. Qu, and R. Johnson, "A nonlinear iterative learning control for robot manipulators in the presence of actuator dynamics," *Internation Journal of Robotics and Automation*, vol. 15, no. 3, pp. 119–130, 2000.

[36] J.-X. Xu, "The frontiers of iterative learning control-ii," *Journal of Systems*, vol. 46, no. 5, pp. 233–243, 2002.

[37] T.-Y. Kuc and W.-G. Han, "An adaptive pid learning control of robot manipulators," *Automatica*, vol. 36, no. 5, pp. 717–725, 2000.

[38] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460–469, 2007.

[39] J. H. Chung and S. A. Velinsky, "Robust interaction control of a mobile manipulator–dynamic model based coordination," *Journal of Intelligent and Robotic Systems*, vol. 26, no. 1, pp. 47–63, 1999.

[40] D. Salima, A. Benali, and F. Abdessemed, "Modelling and contol of an omni-directional mobile manipulator," *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 3, pp. 601–616, 2012.

[41] V. Padois, J.-Y. Fourquet, and P. Chiron, "Kinematic and dynamic model-based control of wheeled mobile manipulators: A unified framework for reactive approaches," *Robotica*, vol. 25, no. 02, pp. 157–173, 2007.

[42] A. De Luca, G. Oriolo, and P. R. Giordano, "Kinematic control of nonholonomic mobile manipulators in the presence of steering wheels," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 1792–1798, IEEE, 2010.

[43] L. Sciavicco and B. Siciliano, "Modelling and control of robot manipulators," 1996.

[44] S. Kucuk and Z. Bingul, *Robot kinematics: Forward and inverse kinematics*. INTECH Open Access Publisher, 2006.

[45] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in MATLAB*. Springer, 2011. ISBN 978-3-642-20143-1.

[46] J. Barraquand and L. Jean-Claude, "On nonholonomic mobile robots and optimal maneuvering," in *In Intelligent Control, 1989. Proceedings., IEEE International Symposium on*, pp. 340–347, IEEE, 1989.

[47] E. S. Conkur and R. Buckingham, "Clarifying the definition of redundancy as used in robotics," *Robotica*, vol. 15, pp. 583–586, 9 1997.

[48] F. Zhu, "Full-order and reduced-order observer-based synchronization for chaotic systems with unknown disturbances and parameters," *Physics letters A*, vol. 372, no. 3, pp. 223–232, 2008.

[49] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.

[50] S. Chiaverini, B. Siciliano, and O. Egeland, "Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator," *Control Systems Technology, IEEE Transactions on*, vol. 2, no. 2, pp. 123–134, 1994.

# Appendix A

# Mathematical Equations

## A.1 Kinematic Modeling of 5-DOF Mobile Manipulator

In 2.2.2, a 2-DOF manipulator armed nonholonomic mobile manipulator and its kinematics is presented in Eqs. (2.13), (2.14) and (2.15). From Eq.(2.15), Jacobian matrix of the kinematics can be calculated. Jacobian of the 5-DOF mobile manipulator is given in terms of linear and angular velocities of the wheeled platform and also in terms of right and left wheel velocities of the mobile platform.

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \\ J_{41} & J_{42} & J_{43} & J_{44} \\ J_{51} & J_{52} & J_{53} & J_{54} \end{bmatrix} \tag{A.1}$$

where

$$J_{11} = \cos\theta_b, J_{12} = 0, J_{13} = 0, J_{14} = 0,$$
$$J_{21} = \sin\theta_b, J_{22} = 0, J_{23} = 0, J_{24} = 0,$$
$$J_{31} = \cos\theta_b$$
$$J_{32} = l_2 \sin\theta_2 \sin(\theta_b + \theta_1)$$
$$J_{33} = l_2 \sin\theta_2 \sin(\theta_b + \theta_1)$$
$$J_{34} = -l_2 \cos\theta_2 \cos(\theta_b + \theta_1)$$
$$J_{41} = \sin\theta_b, \quad J_{42} = -l_2 \sin\theta_2 \cos(\theta_b + \theta_1)$$
$$J_{43} = -l_2 \sin\theta_2 \cos(\theta_b + \theta_1), \quad J_{44} = -l_2 \cos\theta_2 \sin(\theta_b + \theta_1)$$
$$J_{51} = 0, J_{52} = 0, J_{53} = 0, J_{54} = L_2 \cos\theta_2$$
$$OR$$

$$
\begin{aligned}
J_{11} &= \frac{R}{2}\cos\theta_b, J_{12} = \frac{R}{2}\cos\theta_b, J_{13} = 0, J_{14} = 0 \\
J_{21} &= \frac{R}{2}\sin\theta_b, J_{22} = \frac{R}{2}\sin\theta_b, J_{23} = 0, J_{24} = 0 \\
J_{31} &= \frac{R}{2}\cos\theta_b - \big[L_2\cos\theta_2\sin(\theta_b+\theta_1\big]\frac{R}{2D} \\
J_{32} &= \frac{R}{2}\cos\theta_b + \big[L_2\cos\theta_2\sin(\theta_b+\theta_1\big]\frac{R}{2D} \\
J_{33} &= -L_2\cos\theta_2\sin(\theta_b+\theta_1), J_{34} = -L_2\sin\theta_2\cos(\theta_b+\theta_1) \\
J_{41} &= \frac{R}{2}\sin\theta_b + \big[L_2\cos\theta_2\cos(\theta_b+\theta_1\big]\frac{R}{2D} \\
J_{42} &= \frac{R}{2}\sin\theta_b - \big[L_2\cos\theta_2\cos(\theta_b+\theta_1\big]\frac{R}{2D} \\
J_{43} &= L_2\cos\theta_2\cos(\theta_b+\theta_1, J_{44} = -L_2\sin\theta_2\sin(\theta_b+\theta_1) \\
J_{51} &= 0, J_{52} = 0, J_{53} = 0, J_{54} = L_2\cos\theta_2
\end{aligned}
$$

## A.2 The Dynamics Model

Let $L_1$ and $L_2$ denote the distance between joints and the center of mass of the links. The coordinate of the center of mass of Link 1 can be written as,

$$
\begin{aligned}
x_1 &= x \\
y_1 &= y \\
z_1 &= L_2
\end{aligned} \tag{A.2}
$$

the coordinate of the center of mass of link $z$ can be obtained as follows

$$
\begin{aligned}
x_2 &= x_1 + L_2\cos\theta_2\cos(\theta_b+\theta_1) \\
y_2 &= y_1 + L_2\cos\theta_2\sin(\theta_b+\theta_1) \\
z_2 &= l_1 + L_2 Sin(\theta_2)
\end{aligned} \tag{A.3}
$$

The total kinetic energy can be written as follow

$$
\begin{aligned}
K(q,\dot{q}) = &\frac{1}{2}m_b(\dot{x}^2+\dot{y}^2) + \frac{1}{2}I_b\dot{\theta}_b^2 + \frac{1}{2}m_1(\dot{x}^2+\dot{y}^2) + \frac{1}{2}I_1(\dot{\theta}_b+\dot{\theta}_1)^2 \\
&+ \frac{1}{2}m_2(\dot{x}^2+\dot{y}^2) + \frac{1}{2}I_2[(\dot{\theta}_b+\dot{\theta}_1)^2+\dot{\theta}_2)^2]^2
\end{aligned} \tag{A.4}
$$

Total potential energy is

$$P(q) = m_1 g l_1 + m_2 g l_1 + L_2 \sin\theta_2. \tag{A.5}$$

The velocity constraints of the $MM$ system is

$$\dot{x}\sin(\psi) - \dot{y}\cos(\psi) = 0. \tag{A.6}$$

From Routh equation, we can obtain

$$\frac{\partial}{\partial t}\Big(\frac{\partial L}{\partial \dot{q}}\Big) - \frac{\partial L}{\partial q} = Q + A(q)^T\lambda, \tag{A.7}$$

where, $L = K - P$

$Q =$ The forces or torques acted on the plate form and links

$\lambda =$ Langrangian multiplier

Substituting Eq. (A.4) and (A.5) into Eq. (A.7), we can get Eq. (A.8).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + f = B(q)\tau \tag{A.8}$$

where

$$M = \begin{bmatrix} M_b & M_{ba} \\ M_{ab} & M_a \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} C_b & C_{ba} \\ C_{ab} & C_a \end{bmatrix}$$

$$G = \begin{bmatrix} G_b \\ G_a \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ m_2 g L_2 Cos(\theta_2) \end{bmatrix}$$

$$B(q) = \begin{bmatrix} B_b & 0 \\ 0 & B_a \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_b \\ \tau_a \end{bmatrix}$$

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} & M_{15} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} \end{bmatrix}$$

$$C = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} \end{bmatrix}$$

$$\begin{aligned}
M_{11} &= M_{22} = m_p + m_1 + m_2 \\
M_{33} &= I_b + I_1 + I_2 + m_2 L_2^2 \cos^2\theta_2 \\
M_{44} &= I_1 + I_2 + m_2 L_2^2 \cos^2\theta_2 \\
M_{55} &= I_2 + m_2 L_2^2 \cos^2\theta_2 \\
M_{12} &= M_{21} = M_{35} = M_{53} = M_{45} = M_{54} = 0 \\
M_{13} &= M_{31} = M_{41} = M_{14} = -m_2 L_2 \cos\theta_2 \sin(\theta_b + \theta_1) \\
M_{15} &= M_{51} = -m_2 L_2 \sin\theta_2 \cos(\theta_b + \theta_1) \\
M_{23} &= M_{32} = M_{24} = M_{42} = m_2 L_2 \cos(\theta_2) \cos(\theta_b + \theta_1) \\
M_{25} &= M_{52} = -m_2 L_2 \sin(\theta_b + \theta_1) \\
M_{34} &= M_{43} = I_1 + I_2 + m_2 L_2^2 \cos^2\theta_2 \\
C_{ij} &= 0, \quad i = 1, 2, 3, 4, 5.\ j = 1, 2 \\
C_{13} &= C_{14} = -m_2 L_2 \cos\theta_2 \cos(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) \\
&\quad + m_2 L_2 \sin\theta_2 \sin(\theta_b + \theta_1)\dot{\theta}_2 \\
C_{15} &= -m_2 L_2 \cos\theta_2 \cos(\theta_b + \theta_1)\dot{\theta}_2 \\
&\quad + m_2 L_2 \sin\theta_2 \sin(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_2) \\
C_{23} &= C_{24} = -m_2 L_2 \cos\theta_2 \sin(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) \\
&\quad + m_3 L_2 \sin\theta_2 \cos(\theta_b + \theta_1)\dot{\theta}_2 \\
C_{25} &= -m_2 L_2 \sin\theta_2 \cos(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1)
\end{aligned}$$

$$
\begin{aligned}
& - m_2 L_2 \cos\theta_2 \sin(\theta_b + \theta_1)\dot{\theta}_2 \\
C_{35} &= C_{45} = m_2 L_2^2 \cos\theta_2 \sin\theta_2 (\dot{\theta}_b + \dot{\theta}_1) \\
C_{33} &= C_{34} = C_{43} = C_{44} = -m_2 L_2^2 \cos\theta_2 \sin\theta_2 \dot{\theta}_2 \\
C_{53} &= C_{54} = m_2 L_2^2 \cos\theta_2 \sin\theta_2 (\dot{\theta}_b + \dot{\theta}_1) \\
C_{55} &= m_2 L_2^2 \cos\theta_2 \sin\theta_2 \dot{\theta}_2 \\
G &= [0 \ 0 \ 0 \ 0 \ m_2 g L_2 \cos\theta_2]^T
\end{aligned}
$$

## A.3 Regressor Matrix

$$Y = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} & Y_{35} & Y_{36} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} & Y_{45} & Y_{46} \end{bmatrix}$$

$$p = \begin{bmatrix} m_t & m_2 L_2 & m_2 L_2^2 & I_b & I_1 & I_2 \end{bmatrix}^T$$

$$Y_{11} = \frac{R^2}{4}(\dot{v} + \dot{\omega}) + \frac{R^2}{4}(\sin^2\theta_b - \cos^2\theta_b)\omega$$

$$Y_{12} = 2\cos\theta_2 \frac{R^2}{4D}(\cos(\theta_b + \theta_1) - \sin(\theta_b + \theta_1)\cos\theta_b)\dot{v} + \frac{R}{2}\cos\theta_2(\cos(\theta_b + \theta_1)\sin\theta_b - \sin(\theta_b + \theta_1)\cos\theta_b))\ddot{\theta}_1 - \frac{R}{2}\sin\theta_2(\cos(\theta_b + \theta_1)\cos\theta_b + \sin(\theta_b + \theta_1)\sin\theta_b)\ddot{\theta}_2 + \frac{R^2}{4D}\cos\theta_2(\cos(\theta_b + \theta_1)\cos\theta_b + \sin(\theta_b + \theta_1)\sin\theta_b)v - \frac{R^2}{4D}\cos\theta_b(\cos\theta_2\cos(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \cos\theta_2\cos(\theta_b + \theta_1)\dot{\theta}_2)v - \frac{R^2}{4D}\sin\theta_b(\cos\theta_2\sin(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \sin\theta_2\cos(\theta_b + \theta_1)\dot{\theta}_2)v + \frac{R^2}{4D}\cos\theta_2(\cos\theta_b\sin(\theta_b + \theta_1) + \sin\theta_b\cos(\theta_b + \theta_1))\omega + \frac{R^2}{4D}\cos\theta_b(\cos\theta_2\cos(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \sin\theta_2\sin(\theta_b + \theta_1)\dot{\theta}_2)\omega + \frac{R^2}{4D}\sin\theta_b(\cos\theta_2\sin(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \sin\theta_2\cos(\theta_b + \theta_1)\dot{\theta}_2)\omega - \frac{R}{2}\cos\theta_b(\cos\theta_2\cos(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \sin\theta_2\sin(\theta_b + \theta_1)\dot{\theta}_2)\dot{\theta}_1 - \frac{R}{2}\sin\theta_b(\cos\theta_2\sin(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \sin\theta_2\cos(\theta_b + \theta_1)\dot{\theta}_2)\dot{\theta}_1 + \frac{R}{2}\cos\theta_b(\sin\theta_2\sin(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \cos\theta_2\cos(\theta_b + \theta_1)\dot{\theta}_2)\dot{\theta}_2 - \frac{R}{2}\sin\theta_b(\sin\theta_2\cos(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1) - \cos\theta_2\sin(\theta_b + \theta_1)\dot{\theta}_2)\dot{\theta}_2$$

$$Y_{13} = \frac{R^2}{4D^2}\cos^2\theta_2\dot{v} - \frac{R^2}{4D^2}\cos^2\theta_2\dot{\omega} + \frac{R}{2D}\cos^2\theta_2\ddot{\theta}_1 - \frac{R^2}{4D^2}\cos\theta_2\sin\theta_2\dot{\theta}_2 v + \frac{R^2}{4D^2}\cos\theta_2\sin\theta_2\dot{\theta}_2\omega - \frac{R}{2D}\cos\theta_2\sin\theta_2\dot{\theta}_2\dot{\theta}_1 - \frac{R}{2D}\cos\theta_2\sin\theta_2(\dot{\theta}_b + \dot{\theta}_1)\dot{\theta}_2$$

$$Y_{14} = \frac{R^2}{4D^2}\dot{v} - \frac{R^2}{4D^2}\dot{\omega}$$

$$Y_{15} = \frac{R^2}{4D^2}\dot{v} - \frac{R^2}{4D^2}\dot{\omega} + \frac{R}{2D}\dot{\theta}_1$$

$$Y_{16} = \frac{R^2}{4D^2}\dot{v} - \frac{R^2}{4D^2}\dot{\omega} + \frac{R}{2D}\dot{\theta}_1$$

$$Y_{21} = \frac{R^2}{4}\dot{v} + \frac{R^2}{4}\dot{\omega} + \frac{R^2}{4}(\sin^2\theta_b - \cos^2\theta_b)\omega$$

$$\begin{aligned}
Y_{22} = {} & 2\cos\theta_2\sin(\theta_b+\theta_1)\frac{R^2}{4D}\cos\theta_b\dot{\omega} - 2\cos\theta_2\cos(\theta_b+\theta_1)\frac{R^2}{4D}\sin\theta_b\dot{\omega} \\
& - \cos\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\ddot{\theta}_1 + \cos\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\ddot{\theta}_1 \\
& - \sin\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\ddot{\theta}_2 + \sin\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\ddot{\theta}_2 \\
& - \cos\theta_2\sin(\theta_b+\theta_1)\frac{R^2}{4D}\sin\theta_b v - \cos\theta_2\cos(\theta_b+\theta_1)\frac{R^2}{4D}\cos\theta_b v \\
& - \cos\theta_2\cos(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R^2}{4D}\cos\theta_b v + \sin\theta_2\sin(\theta_b+\theta_1)\dot{\theta}_2\frac{R^2}{4D}\cos\theta_b v \\
& - \cos\theta_2\sin(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R^2}{4D}\sin\theta_b v + \sin\theta_2\cos(\theta_b+\theta_1)\dot{\theta}_2\frac{R^2}{4D}\sin\theta_b v \\
& - \cos\theta_2\sin(\theta_b+\theta_1)\frac{R^2}{4D}\cos\theta_b\omega - \cos\theta_2\cos(\theta_b+\theta_1)\frac{R^2}{4D}\sin\theta_b\omega \\
& + \cos\theta_2\cos(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R^2}{4D}\cos\theta_b\omega - \sin\theta_2\sin(\theta_b+\theta_1)\dot{\theta}_2\frac{R^2}{4D}\cos\theta_b\omega \\
& + \cos\theta_2\sin(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R^2}{4D}\sin\theta_b\omega - \sin\theta_2\cos(\theta_b+\theta_1)\dot{\theta}_2\frac{R^2}{4D}\sin\theta_b\omega \\
& - \cos\theta_2\cos(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R}{2}\cos\theta_b\dot{\theta}_1 + \sin\theta_2\sin(\theta_b+\theta_1)\dot{\theta}_2\frac{R}{2}\cos\theta_b\dot{\theta}_1 \\
& - \cos\theta_2\sin(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R}{2}\sin\theta_b\dot{\theta}_1 + \sin\theta_2\cos(\theta_b+\theta_1)\dot{\theta}_2\frac{R}{2}\sin\theta_b\dot{\theta}_1 \\
& + \sin\theta_2\sin(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R}{2}\cos\theta_b\dot{\theta}_2 - \cos\theta_2\cos(\theta_b+\theta_1)\dot{\theta}_2\frac{R}{2}\cos\theta_b\dot{\theta}_2 \\
& - \sin\theta_2\cos(\theta_b+\theta_1)(\dot{\theta}_b+\dot{\theta}_1)\frac{R}{2}\sin\theta_b\dot{\theta}_2 - \cos\theta_2\sin(\theta_b+\theta_1)\dot{\theta}_2\frac{R}{2}\sin\theta_b\dot{\theta}_2
\end{aligned}$$

$$\begin{aligned}
Y_{23} = {} & -\frac{R^2}{4D^2}\cos^2\theta_2\dot{v} + \frac{R^2}{4D^2}\cos^2\theta_2\dot{\omega} - \frac{R^2}{4D^2}\cos^2\theta_2\ddot{\theta}_1 + \cos\theta_2\sin\theta_2\dot{\theta}_2\frac{R^2}{4D^2}v \\
& - \cos\theta_2\sin\theta_2\dot{\theta}_2\frac{R^2}{4D^2}\omega + \cos\theta_2\sin\theta_2\dot{\theta}_2\frac{R}{2D}\dot{\theta}_1 + \cos\theta_2\sin\theta_2(\dot{\theta}_b+\dot{\theta}_1)\frac{R}{2D}\dot{\theta}_2
\end{aligned}$$

$$Y_{24} = -\frac{R^2}{4D^2}\cos^2\theta_2\dot{v} + \frac{R^2}{4D^2}\cos^2\theta_2\dot{\omega}$$

$$Y_{25} = -\frac{R^2}{4D^2}\cos^2\theta_2\dot{v} + \frac{R^2}{4D^2}\cos^2\theta_2\dot{\omega} + \frac{R}{2D}\ddot{\theta}_1$$

$$Y_{26} = -\frac{R^2}{4D^2}\cos^2\theta_2\dot{v} + \frac{R^2}{4D^2}\cos^2\theta_2\dot{\omega} + \frac{R}{2D}\ddot{\theta}_1$$

$$Y_{31} = 0$$

$$\begin{aligned}
Y_{32} = {} & -\cos\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\dot{v} + \cos\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\dot{v} - \cos\theta_2 \\
& \sin(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\dot{\omega} + \cos\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\dot{\omega} + \cos\theta_2\sin(\theta_b+\theta_1)
\end{aligned}$$

$$\frac{R}{2}\sin\theta_b v + \cos\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b v + \cos\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\omega$$
$$+\cos\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\omega$$

$$Y_{33} = \frac{R}{2D}\cos^2\theta_2\dot{v} - \frac{R}{2D}\cos^2\theta_2\dot{\omega} + \cos^2\theta_2\ddot{\theta}_1 - \cos\theta_2\sin\theta_2\dot{\theta}_2\frac{R}{2D}v + \cos\theta_2$$
$$\sin\theta_2\dot{\theta}_2\frac{R}{2D}\omega - \cos\theta_2\sin\theta_2\dot{\theta}_2\dot{\theta}_1 + \cos\theta_2\sin\theta_2(\dot{\theta}_b+\dot{\theta}_1)\dot{\theta}_2$$

$$Y_{34} = 0$$

$$Y_{35} = \frac{R}{2D}\dot{v} - \frac{R}{2D}\dot{\omega} + \ddot{\theta}_1$$

$$Y_{36} = \frac{R}{2D}\dot{v} - \frac{R}{2D}\dot{\omega} + \ddot{\theta}_1$$

$$Y_{41} = 0$$

$$Y_{42} = \sin\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\dot{v} - \sin\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\dot{v} - \sin\theta_2$$
$$\cos(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\dot{\omega} - \sin\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\dot{\omega} + \sin\theta_2\cos(\theta_b+\theta_1)$$
$$\frac{R}{2}\sin\theta_b v + \sin\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b v + \sin\theta_2\cos(\theta_b+\theta_1)\frac{R}{2}\cos\theta_b\omega$$
$$+\sin\theta_2\sin(\theta_b+\theta_1)\frac{R}{2}\sin\theta_b\omega$$

$$Y_{43} = \cos\theta_2\sin\theta_2(\dot{\theta}_b+\dot{\theta}_1)\frac{R}{2D}v - \cos\theta_2\sin\theta_2(\dot{\theta}_b+\dot{\theta}_1)\frac{R}{2D}\omega + \cos\theta_2\sin\theta_2$$
$$(\dot{\theta}_b+\dot{\theta}_1)\dot{\theta}_1 + \cos\theta_2\sin\theta_2\dot{\theta}_2\dot{\theta}_2$$

$$Y_{44} = 0$$

$$Y_{45} = 0$$

$$Y_{46} = 0$$

### A.3.1 Regressor Matrix Y in Terms of Vehicle Linear and Angular Velocities

$$Y = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} & Y_{17} & Y_{18} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} & Y_{27} & Y_{28} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} & Y_{35} & Y_{36} & Y_{37} & Y_{38} \\ Y_{41} & Y_{42} & Y_{43} & Y_{44} & Y_{45} & Y_{46} & Y_{47} & Y_{48} \end{bmatrix} \tag{A.9}$$

The parameters are:

$$p = \begin{bmatrix} m_b & m_1 & m_2 & m_2 L_2 & m_2 L_2^2 & I_b & I_1 & I_2 \end{bmatrix}^T \tag{A.10}$$

$Y_{11} = Y_{12} = Y_{13} = \dot{\eta}_1$

$Y_{14} = \cos\theta_2(\cos(\theta_b + \theta_1)\sin\theta_b - \sin(\theta_b + \theta_1)\cos\theta_b)\dot{\eta}_2 + \cos\theta_2\sin(\theta_b + \theta_1)$
$(\sin\theta_b - \cos\theta_b)\dot{\eta}_3 - \sin\theta_2(\cos(\theta_b + \theta_1)\cos\theta_b + \sin(\theta_b + \theta_1)\sin\theta_b)\dot{\eta}_4 + \cos\theta_b$
$(\sin(\theta_b + \theta_1)\sin\theta_2\dot{\theta}_2 - \cos(\theta_b + \theta_1)\cos\theta_2(\dot{\theta}_b + \dot{\theta}_1))(\eta_2 + \eta_3) + \sin\theta_b(\cos(\theta_b + \theta_1)$
$\sin\theta_2\dot{\theta}_2 - \cos\theta_2\cos(\theta_b + \theta_1)(\dot{\theta}_b + \dot{\theta}_1))(\eta_2 + \eta_3) + \cos\theta_b(\sin(\theta_b + \theta_1)\sin\theta_2$
$(\dot{\theta}_b + \dot{\theta}_1) - \cos\theta_2\cos(\theta_b + \theta_1)\dot{\theta}_2)\eta_4 + \sin\theta_b(\cos(\theta_b + \theta_1)\sin\theta_2(\dot{\theta}_b + \dot{\theta}_1) - \sin\theta_2$
$\cos(\theta_b + \theta_1)\dot{\theta}_2)\eta_4$

$Y_{15} = Y_{16} = Y_{17} = Y_{18} = 0$

$Y_{21} = Y_{22} = Y_{23} = 0$

$Y_{24} = \cos\theta_2(\cos(\theta_b + \theta_1)\sin\theta_b - \sin(\theta_b + \theta_1)\cos\theta_b)\dot{\eta}_1 + \cos\theta_2(\sin(\theta_b + \theta_1)\sin\theta_b + \cos(\theta_b$
$+ \theta_1)\cos\theta_b)\dot{\theta}_b\eta_1$

$Y_{25} = \cos^2\theta_2\dot{\eta}_2 + \cos^2\theta_2\dot{\eta}_3 + \cos\theta_2\sin\theta_2\dot{\theta}_2(\dot{\eta}_2 + \dot{\eta}_3) + \cos\theta_2\sin\theta_2(\dot{\theta}_b + \dot{\theta}_1)\dot{\eta}_4$

$Y_{26} = \eta_2$

$Y_{27} = Y_{28} = \dot{\eta}_2 + \dot{\eta}_3$

$Y_{31} = Y_{32} = Y_{33} = Y_{36} = 0$

$Y_{34} = \cos\theta_2\sin(\theta_b + \theta_1)(\sin\theta_b - \cos\theta_b)\dot{v} + \cos\theta_2\sin(\theta_b + \theta_1)(\sin\theta_b + \cos\theta_b)\dot{\theta}_2 v$

$Y_{35} = \cos^2\theta_2\dot{\eta}_2 + \cos^2\theta_2\dot{\eta}_3 - \cos\theta_2\sin\theta_2\dot{\theta}_2(\dot{\eta}_2 + \dot{\eta}_3) - \cos\theta_2\sin\theta_2(\dot{\theta}_b + \dot{\theta}_1)\dot{\eta}_4$

$Y_{36} = 0$

$Y_{37} = Y_{38} = \dot{\eta}_2 + \dot{\eta}_3$

$$Y_{41} = Y_{42} = Y_{43} = 0$$

$$Y_{44} = -\sin\theta_2 \cos(\theta_b + \theta_1)(\cos\theta_b - \sin\theta_b)\dot{v} + \sin\theta_2 \cos(\theta_b + \theta_1)(\sin\theta_b + \cos\theta_b)\dot{\theta}_b v$$

$$Y_{45} = \cos^2\theta_2\dot{\eta}_2 + \cos^2\theta_2\dot{\eta}_3 + \cos\theta_2 \sin\theta_2\dot{\theta}_2(\dot{\eta}_2 + \dot{\eta}_3) + \cos\theta_2 \sin\theta_2(\dot{\theta}_b + \dot{\theta}_1)\dot{\eta}_4$$

$$Y_{46} = Y_{47} = 0$$

$$Y_{48} = \dot{\eta}_4$$

## A.4 MBC control Dynamics

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)\tau + f$$

$$q_v = [x, y, \theta]^T \quad q_a = [\theta_1, \theta_2]^T \quad q = [q_v, q_a]^T$$

$$A = [\cos\theta, \sin\theta, 0]^T$$

$$M_{v11} = \begin{bmatrix} m_{p12} + \dfrac{2I_w \sin^2\theta}{r^2} & -\dfrac{2I_w}{r^2}\sin\theta\cos\theta \\ \dfrac{2I_w}{r^2}\sin\theta\cos\theta & m_{p12} + \dfrac{2I_w\cos^2\theta}{r^2} \end{bmatrix}$$

$$M_{v12} = \begin{bmatrix} -m_{12}d\sin\theta \\ m_{12}d\cos\theta \end{bmatrix}$$

$$M_{v21} = M_{v12}^T \quad M_{v22} = M_{11}^1$$

$$C_v = \begin{bmatrix} \dfrac{2I_w}{r^2}\dot{\theta}\sin\theta\cos\theta & -\dfrac{2I_w}{r^2}\dot{\theta}\cos^2\theta & 0 \\ \dfrac{2I_w}{r^2}\dot{\theta}\sin^2\theta & -\dfrac{2I_w}{r^2}\dot{\theta}\sin\theta\cos\theta & 0 \\ -m_{12}d\dot{\theta}\cos\theta & -m_{12}d\dot{\theta}\cos\theta & 0 \end{bmatrix}$$

$$M_{11}^1 = I_p + I_{12} + m12d^2 + \frac{2I_w l^2}{r^2}$$

$$M_a = diag[I_{12}, I_2]$$

$$m_{p12} = m_p + m_{12} \quad m_{12} = m1 + m2$$

$$I_{12} = I_1 + I_2$$

$$M_{va} = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \\ I_{12} & 0.0 \end{bmatrix}$$

$$B = \begin{bmatrix} \dfrac{\sin\theta}{r} & -\dfrac{\sin\theta}{r} & 0.0 & 0.0 \\ \dfrac{-\cos\theta}{r} & \dfrac{\cos\theta}{r} & 0.0 & 0.0 \\ -\dfrac{l}{r} & \dfrac{l}{r} & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

$$C_{va} = 0.0 \quad C_{av} = C_{va}^T \quad C_a = 0.0$$

$$G_v = [0.0, 0.0, 0.0]^T \quad G_a = [0.0, m_2 g l_2 \sin\theta_2]^T$$

$$H = \begin{bmatrix} -\tan\theta & 0.0 \\ 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$$

$$\tau_v = [\tau_L, \tau_r]^T \quad \tau_a = [\tau_1, \tau_2]^T$$

$$\zeta = [\zeta_1 \zeta_2 \zeta_3]^T \quad \dot{\zeta} = [v, \omega, \theta_1]^T$$

## A.5 MATLAB Codes

Mobile Manipulator System Block

```
function etaD = SYSTEM(Taubar, eta, q, thetabD, theta1D,theta2D)
%#codegen


mb = 32; %32
m1 = 3;  %3
m2 = 3;  %3

Ib = 1;
I1 = 1;
I2 = 1;

g = 9.8;

D = 1;
L1 = 1;
L2 = 1;
R = 0.5;

% thetab = orientation of the mobile manipulator
% q = [x; y; thetab; tehta1; theat2];

thetab = q(3);
theta1 = q(4);
theta2 = q(5);


%% Matrix M
% M =[M11 M12 M13 M14 M15;
%     M21 M22 M23 M24 M25;
%     M31 M32 M33 M34 M35;
%     M41 M42 M43 M44 M45;
%     M51 M52 M53 M54 M55]
```

```
M11= mb + m1 + m2;
M12= 0;
M13= -m2*L2*cos(theta2)*sin(thetab+theta1);
M14= -m2*L2*cos(theta2)*sin(thetab+theta1);
M15= -m2*L2*sin(theta2)*cos(thetab+theta1);

M21= 0;
M22= mb + m1 + m2;
M23= m2*L2*cos(theta2)*cos(thetab+theta1);
M24= m2*L2*cos(theta2)*sin(thetab+theta1);
M25= -m2*L2*sin(theta2)*sin(thetab+theta1);

M31=-m2*L2*cos(theta2)*sin(thetab+theta1);
M32= m2*L2*cos(theta2)*cos(thetab+theta1);
M33= Ib+I1+I2+(m2*(L2^2)*(cos(theta2)^2));
M34= I1+I2+(m2*(L2^2)*(cos(theta2)^2));
M35= 0;

M41= -m2*L2*cos(theta2)*sin(thetab+theta1);
M42= m2*L2*cos(theta2)*sin(thetab+theta1);
M43= I1+I2+(m2*(L2^2)*(cos(theta2)^2));
M44= I1+I2+(m2*(L2^2)*(cos(theta2)^2));
M45= 0;

M51= -m2*L2*sin(theta2)*cos(thetab+theta1);
M52= -m2*L2*sin(thetab+theta1);
M53= 0;
M54= 0;
M55= I2+(m2*(L2^2)*(cos(theta2)^2));

M =[M11 M12 M13 M14 M15;
M21 M22 M23 M24 M25;
M31 M32 M33 M34 M35;
M41 M42 M43 M44 M45;
M51 M52 M53 M54 M55];

%% Matrix C
%C = [C11 C12 C13 C14 C15;
```

```
%     C21 C22 C23 C24 C25;
%     C31 C32 C33 C34 C35;
%     C41 C42 C43 C44 C45;
%     C51 C52 C53 C54 C55];

C11 = 0;
C12 = 0;
C13 = (-m2*L2*cos(theta2)*cos(thetab+theta1)*(thetabD+theta1D))...
    + m2*L2*sin(theta2)*sin(thetab+theta1)*theta2D;
C14 = (-m2*L2*cos(theta2)*cos(thetab+theta1)*(thetabD+theta1D))...
    + m2*L2*sin(theta2)*sin(thetab+theta1)*theta2D;
C15 = (-m2*L2*cos(theta2)*cos(thetab+theta1)*(theta2D))...
    +( m2*L2*sin(theta2)*sin(thetab+theta1)*(thetabD+theta1D));

C21= 0;
C22= 0;
C23= (-m2*L2*cos(theta2)*sin(thetab+theta1)*(thetabD+theta1D))...
    + m2*L2*sin(theta2)*cos(thetab+theta1)*theta2D;
C24= (-m2*L2*cos(theta2)*sin(thetab+theta1)*(thetabD+theta1D))...
    + m2*L2*sin(theta2)*cos(thetab+theta1)*theta2D;
C25= (-m2*L2*sin(theta2)*cos(thetab+theta1)*(thetabD+theta1D))...
    + m2*L2*cos(theta2)*sin(thetab+theta1)*theta2D;

C31 =0;
C32 =0;
C33 = -m2*L2^2*cos(theta2)*sin(theta2)*theta2D;
C34 = -m2*L2^2*cos(theta2)*sin(theta2)*theta2D;
C35 = -m2*L2^2*cos(theta2)*sin(theta2)*(thetabD+theta1D);

C41 = 0;
C42 = 0;
C43 = -m2*L2^2*cos(theta2)*sin(theta2)*theta2D;
C44 = -m2*L2^2*cos(theta2)*sin(theta2)*theta2D;
C45 = -m2*L2^2*cos(theta2)*sin(theta2)*(thetabD+theta1D);

C51= 0;
C52 =0;
C53 = m2*L2^2*cos(theta2)*sin(theta2)*(thetabD+theta1D);
C54 = m2*L2^2*cos(theta2)*sin(theta2)*(thetabD+theta1D);
```

```
C55 = m2*L2^2*cos(theta2)*sin(theta2)*theta2D;

C = [C11 C12 C13 C14 C15;
C21 C22 C23 C24 C25;
C31 C32 C33 C34 C35;
C41 C42 C43 C44 C45;
C51 C52 C53 C54 C55];

%% G matrix

%G = [G1; G2; G3; G4; G5];
G1= 0;
G2= 0;
G3 =0;
G4 =0;
G5= m2*g*L2*cos(theta2);

G = [G1; G2; G3; G4; G5];

%% E Matrix

%   E = [(1/R)*cos(theta) (1/R)*cos(theta) 0 0;
%        (1/R)*sin(theta) (1/R)*sin(theta) 0 0;
 %        D/R -D/R 0 0;
 %        0 0 1 0;
  %       0 0 0 1];

%% Including Non-holonomic Constraints to the system.

H = [(R/2)*cos(q(3)) (R/2)*cos(q(3)) 0 0;...
    (R/2)*sin(q(3)) (R/2)*sin(q(3)) 0 0;...
    R/2*D -R/2*D 0 0;...
    0 0 1 0;...
    0 0 0 1];
% taking transpose of H
Ht = H';
% taking H/dot
HD = [-(R/2)*sin(q(3))*thetabD -(R/2)*sin(q(3))*thetabD 0 0;...
    (R/2)*cos(q(3))*thetabD (R/2)*cos(q(3))*thetabD 0 0;...
```

```
    0 0 0 0; 0 0 0 0; 0 0 0 0];
%% Dynamics can be expressed by Mbar, Cbar, Gbar

Mbar = Ht*M*H;
Cbar = Ht*(M*HD+C*H);
Gbar = Ht*G;

etaD =(Mbar)\(Taubar- Cbar*eta - Gbar);
```

Kinematic Transformation and I.K. Block

```
function [qD, Deltainverse, Sidot, DeltainverseDot, PP, XX]  = KT(eta,q)
%#codegen

%KT = kinematic Transformation
R = 0.5;
D = 1;
L2 = 1; %length of the second link
theta2 = q(5);
theta1 = q(4);
thetab = q(3);

%% Including Non-holonomic Constraints to the system.

H = [(R/2)*cos(q(3)) (R/2)*cos(q(3)) 0 0;...
    (R/2)*sin(q(3)) (R/2)*sin(q(3)) 0 0;...
    R/2*D -R/2*D 0 0;...
    0 0 1 0;...
    0 0 0 1];



%%
%qdot
qD = H*eta;

ThetabDot = qD(3);
Theta1Dot = qD(4);
```

```
Theta2Dot = qD(5);


%Jh1 = -L2*cos(theta2)*sin(theta1+thetab);
%Jh2 = -L2*sin(theta2)*cos(theta1+thetab);
%Jh3 =  L2*cos(theta2)*cos(theta1+thetab);
%Jh4 = -L2*sin(theta2)*sin(theta1+thetab);
%Jh5 = L2*cos(theta2);

%Jh1 = L2*sin(theta2)*sin(theta1+thetab);
%Jh2 = -L2*cos(theta2)*cos(theta1+thetab);
%Jh3 = -L2*sin(theta2)*cos(theta1+thetab);
%Jh4 =-L2*cos(theta2)*sin(theta1+thetab);
%Jh5 = L2*sin(theta2);

J11 = (R/2)*cos(thetab);
J12 = (R/2)*cos(thetab);
J13 = 0;
J14 = 0;

J21 = (R/2)*sin(thetab);
J22 = (R/2)*sin(thetab);
J23 = 0;
J24 = 0;

J31 = (R/2)*cos(thetab) - (L2*cos(theta2)*sin(theta1+thetab))*(R/2*D);
J32 = (R/2)*cos(thetab) + (L2*cos(theta2)*sin(theta1+thetab))*(R/2*D);
J33 = -L2*cos(theta2)*sin(theta1+thetab);
J34 = -L2*sin(theta2)*cos(theta1+thetab);

J41 = (R/2)*sin(thetab) + (L2*cos(theta2)*cos(theta1+thetab))*(R/2*D);
J42 = (R/2)*sin(thetab) - (L2*cos(theta2)*cos(theta1+thetab))*(R/2*D);
J43 = L2*cos(theta2)*cos(theta1+thetab);
J44 = -L2*sin(theta2)*sin(theta1+thetab);

J51 = 0;
J52 = 0;
J53 = 0;
J54 = L2*cos(theta2);
```

```
J = [J11 J12 J13 J14;
     J21 J22 J23 J24;
     J31 J32 J33 J34;
     J41 J42 J43 J44;
     J51 J52 J53 J54];

%Delta = [cos(thetab) 0 0 0; sin(thetab) 0 0 0;
%         cos(thetab) Jh1 Jh1 Jh2;sin(thetab) Jh3 Jh3 Jh4; 0 0 0 Jh5 ];

%sidot
Sidot = J*eta;


J11d = -(R/2)*sin(thetab)*ThetabDot;
J12d = -(R/2)*sin(thetab)*ThetabDot;
J13d = 0;
J14d = 0;

J21d = (R/2)*cos(thetab)*ThetabDot;
J22d = (R/2)*cos(thetab)*ThetabDot;
J23d = 0;
J24d = 0;

J31d = -(R/2)*sin(thetab)*ThetabDot...
    - (L2*cos(theta2)*cos(theta1+thetab)*(ThetabDot+Theta1Dot)...
    - L2*sin(theta2)*sin(theta1+thetab)*Theta2Dot)*(R/2*D);
J32d = -(R/2)*sin(thetab)*ThetabDot...
    + (L2*cos(theta2)*cos(theta1+thetab)*(ThetabDot+Theta1Dot)...
    - L2*sin(theta2)*sin(theta1+thetab)*Theta2Dot)*(R/2*D);
J33d = -L2*cos(theta2)*cos(theta1+thetab)*(ThetabDot+Theta1Dot)...
    + L2*sin(theta2)*sin(theta1+thetab)*Theta2Dot;
J34d = L2*sin(theta2)*sin(theta1+thetab)*(ThetabDot+Theta1Dot)...
    + L2*cos(theta2)*cos(theta1+thetab)*Theta2Dot;

J41d = (R/2)*cos(thetab)*ThetabDot...
    + (-L2*cos(theta2)*sin(theta1+thetab)*(ThetabDot+Theta1Dot)...
    - L2*sin(theta2)*cos(theta1+thetab)*Theta2Dot)*(R/2*D);
J42d = (R/2)*cos(thetab)*ThetabDot...
```

```
    - (-L2*cos(theta2)*sin(theta1+thetab)*(ThetabDot+Theta1Dot)...
    - L2*sin(theta2)*cos(theta1+thetab)*Theta2Dot)*(R/2*D);
J43d = -L2*cos(theta2)*sin(theta1+thetab)*(ThetabDot+Theta1Dot)...
    - L2*sin(theta2)*cos(theta1+thetab)*Theta2Dot;
J44d = -L2*sin(theta2)*cos(theta1+thetab)*(ThetabDot+Theta1Dot)...
    - L2*cos(theta2)*sin(theta1+thetab)*Theta2Dot;

J51d = 0;
J52d = 0;
J53d = 0;
J54d = -L2*sin(theta2)*Theta2Dot;

Jd = [J11d J12d J13d J14d;
     J21d J22d J23d J24d;
     J31d J32d J33d J34d;
     J41d J42d J43d J44d;
     J51d J52d J53d J54d];



%DeltaInverse
%k= 1.5;
k= 1.01;
K2I = [k^2 0 0 0 0; 0 k^2 0 0 0; 0 0 k^2 0 0; 0 0 0 k^2 0; 0 0 0 0 k^2];

Deltainverse = J'/(J*J' + K2I);

% DeltaDot for Delta inverse dot; rule:
%deltainversedot = -deltainverse*deltadot*Deltadot
%DeltaDot = [-sin(thetab)*ThetabDot 0 0 0;
%           cos(thetab)*ThetabDot 0 0 0;
%           -sin(thetab)*ThetabDot Jh1D Jh1D Jh2D;
%           cos(thetab)*ThetabDot Jh3D Jh3D Jh4D; 0 0 0 Jh5D];

DeltainverseDot = -Deltainverse*Jd*Deltainverse;

%For etaCDot

PPP = Deltainverse*J;
PP = eye(4)-PPP;
```

```
xxx1 = -Deltainverse*Jd;
xxx2 = -DeltainverseDot*J;

XX = xxx1+xxx2;
end
```

Regressor Matrix Calculation

```
function Y = fcn(etaC, etaCDot, q, qD)
%#codegen

R = 0.5;
D = 1;
g= 9.8;


vdot = etaCDot(1);
wdot = etaCDot(2);
theta1DD = etaCDot(3);
theta2DD = etaCDot(4);

v = etaC(1);
w = etaC(2);
Theta1D = etaC(3);
Theta2D = etaC(4);

thetab = q(3);
theta1 = q(4);
theta2 = q(5);

thetabD = qD(3);
theta1D = qD(4);
theta2D = qD(5);

Y11 = (R^2/4)*(vdot + wdot) + (R^2/4)*(sin(thetab)^2-cos(thetab)^2)*w ;
Y12 = 2*cos(theta2)*(R^2/4*D)*(cos(thetab+theta1)*sin(thetab)...
        - sin(thetab+theta1)*cos(thetab))*vdot...
```

```
       + (R/2)*cos(theta2)*(cos(thetab+theta1)*sin(thetab)...
       - sin(thetab+theta1)*cos(thetab))*theta1DD...
       - (R/2)*sin(theta2)*(cos(thetab+theta1)*cos(thetab)...
       + sin(thetab+theta1)*sin(thetab))*theta2DD ...
       + (R^2/4*D)*cos(theta2)*(cos(thetab+theta1)*cos(thetab)...
       + sin(thetab+theta1)*sin(thetab))*v...
       - (R^2/4*D)*(thetabD+theta1D)*cos(theta2)*(cos(thetab+theta1)...
       *cos(thetab)...
       + sin(thetab+theta1)*sin(thetab))*v...
       + (R^2/4*D)*sin(theta2)*theta2D*(cos(thetab+theta1)*sin(thetab)...
       + sin(thetab+theta1)*cos(thetab))*v...
       + (R^2/4*D)*cos(theta2)*(cos(thetab+theta1)*sin(thetab)...
       + sin(thetab+theta1)*cos(thetab))*w...
       + (R^2/4*D)*cos(theta2)*(thetabD+theta1D)*(cos(thetab+theta1)...
       *cos(thetab)...
       + sin(thetab+theta1)*sin(thetab))*w...
       - (R^2/4*D)*sin(theta2)*theta2D*(cos(thetab+theta1)*sin(thetab)...
       + sin(thetab+theta1)*cos(thetab))*w...
       - (R/2)*(thetabD+theta1D)*cos(theta2)*(cos(thetab+theta1)...
       *cos(thetab)...
       + sin(thetab+theta1)*sin(thetab))*Theta1D...
       + (R/2)*(theta2D)*sin(theta2)*(sin(thetab+theta1)*cos(thetab)...
       + cos(thetab+theta1)*sin(thetab))*Theta1D...
       + (R/2)*(thetabD+theta1D)*sin(theta2)*(sin(thetab+theta1)...
       *cos(thetab)...
       - cos(thetab+theta1)*sin(thetab))*Theta2D...
       - (R/2)*(theta2D)*cos(theta2)*(cos(thetab+theta1)...
       *cos(thetab)...
       + sin(thetab+theta1)*sin(thetab))*Theta2D ;
Y13 = (R^2/4*D^2)*cos(theta2)^2*(vdot-wdot)...
    + (R/2*D)*cos(theta2)^2* theta1DD...
    + (R^2/4*D^2)* cos(theta2)*sin(theta2)*theta2D*(w-v)...
    - (R/2*D)*cos(theta2)*sin(theta2)*(theta2D*Theta1D...
    + (thetabD+theta1D)*Theta2D);
Y14 = (R^2/4*D^2)*(vdot - wdot);
Y15 = Y14 + (R/2*D)*theta1DD ;
Y16 = Y15;

Y21 = Y11 ;
```

```
Y22 = 2*cos(theta2)*(R^2/4*D)*(-cos(thetab+theta1)*sin(thetab)...
    + sin(thetab+theta1)*cos(thetab))*wdot...
    + (R/2)*cos(theta2)*(cos(thetab+theta1)*sin(thetab)...
    - sin(thetab+theta1)*cos(thetab))*theta1DD...
    - (R/2)*sin(theta2)*(cos(thetab+theta1)*cos(thetab)...
    + sin(thetab+theta1)*sin(thetab))*theta2DD...
    -(R^2/4*D)*cos(theta2)*(cos(thetab+theta1)*cos(thetab)...
    + sin(thetab+theta1)*sin(thetab))*v...
    - (R^2/4*D)*(thetabD+theta1D)*cos(theta2)*(cos(thetab+theta1)...
    *cos(thetab)...
    + sin(thetab+theta1)*sin(thetab))*v...
    + (R^2/4*D)*sin(theta2)*theta2D*(cos(thetab+theta1)*sin(thetab)...
    +sin(thetab+theta1)*cos(thetab))*v...
    - (R^2/4*D)*cos(theta2)*(cos(thetab+theta1)*sin(thetab)...
    + sin(thetab+theta1)*cos(thetab))*w...
    + (R^2/4*D)*cos(theta2)*(thetabD+theta1D)*(cos(thetab+theta1)...
    *cos(thetab)...
    + sin(thetab+theta1)*sin(thetab))*w...
    - (R^2/4*D)*sin(theta2)*theta2D*(cos(thetab+theta1)*sin(thetab)...
    + sin(thetab+theta1)*cos(thetab))*w...
    - (R/2)*(thetabD+theta1D)*cos(theta2)*(cos(thetab+theta1)...
    *cos(thetab)...
    + sin(thetab+theta1)*sin(thetab))*Theta1D...
    + (R/2)*(theta2D)*sin(theta2)*(sin(thetab+theta1)*cos(thetab)...
    +cos(thetab+theta1)*sin(thetab))*Theta1D...
    + (R/2)*(thetabD+theta1D)*sin(theta2)*(sin(thetab+theta1)...
    *cos(thetab)...
    - cos(thetab+theta1)*sin(thetab))*Theta2D...
    - (R/2)*(theta2D)*cos(theta2)*(cos(thetab+theta1)*cos(thetab)...
    + sin(thetab+theta1)*sin(thetab))*Theta2D ;
Y23 = (R^2/4*D^2)*cos(theta2)^2*(wdot-vdot)...
    - (R/2*D)*cos(theta2)^2* theta1DD...
    + (R^2/4*D^2)* cos(theta2)*sin(theta2)*theta2D*(v-w)...
    + (R/2*D)*cos(theta2)*sin(theta2)*(theta2D*Theta1D...
    + (thetabD+theta1D)*Theta2D);
Y24 = (R^2/4*D^2)*(wdot - vdot);
Y25 = Y24 + (R/2*D)*theta1DD ;
Y26 = Y25;
```

```
Y31 = 0;
Y32 = -(R/2)*cos(theta2)*sin(thetab+theta1)*cos(thetab)*(vdot+wdot)...
    + (R/2)*cos(theta2)*cos(thetab+theta1)*sin(thetab)*(vdot+wdot)...
    + (cos(theta2)*sin(thetab+theta1)*sin(thetab)...
    + cos(theta2)*cos(thetab+theta1)*cos(thetab))*(R/2)*v...
    + (R/2)*cos(theta2)*(sin(thetab+theta1)*cos(thetab)...
    +cos(thetab+theta1)*sin(thetab))*w ;
Y33 = (R/2*D)*cos(theta2)^2*(vdot-wdot)...
    + cos(theta2)^2*theta1DD...
    + (R/2*D)*cos(theta2)*sin(theta2)*theta2D*(w-v)...
    + cos(theta2)*sin(theta2)*((thetabD+theta1D)*Theta2D...
    - theta2D*Theta1D);
Y34 = 0;
Y35 = (R/2*D)*(vdot-wdot)+ theta1DD;
Y36 = Y35;

Y41 = 0;
Y42 = (R/2)*sin(theta2)*cos(thetab+theta1)*cos(thetab)*(vdot-wdot)...
    - (R/2)*sin(theta2)*sin(thetab+theta1)*sin(thetab)*(vdot+wdot)...
    + (R/2)*sin(theta2)*(cos(thetab+theta1)*sin(thetab)...
    + sin(thetab+theta1)*cos(thetab))*v...
    + (R/2)*sin(theta2)*(cos(thetab+theta1)*cos(thetab)...
    +sin(thetab+theta1)*sin(thetab))*w + g*cos(theta2);
Y43 = cos(theta2)*sin(theta2)*(thetabD+theta1D)*(R/2*D)*(v-w)...
    + cos(theta2)*sin(theta2)*((thetabD+theta1D)*Theta1D+...
     theta2D*Theta2D);
Y44 = 0;
Y45 = 0;
Y46 = 0;

Y = [Y11 Y12 Y13 Y14 Y15 Y16;
    Y21 Y22 Y23 Y24 Y25 Y26;
    Y31 Y32 Y33 Y34 Y35 Y36;
    Y41 Y42 Y43 Y44 Y45 Y46];

end
```