2020

Rapid Orbital Motion Emulator (ROME): Kinematics Modeling and Control

Ahmed Elsadek Ahmed Seleit
University of Central Florida

RAPID ORBITAL MOTION EMULATOR (ROME): KINEMATICS MODELING AND CONTROL

by

AHMED ELSADEK SELEIT
M.S. CAIRO UNIVERSITY, 2016

A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science
in the Department of Mechanical and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2020

Major Professor: Tarek A. Elgohary

ABSTRACT

Space missions design requires already tested and trusted control algorithms for spacecraft motion. Rapidly testing control algorithms at a low cost is essential. A novel robotic system that emulates orbital motion in a laboratory environment is presented. The system is composed of a six degree of freedom robotic manipulator fixed on top of an omnidirectional ground vehicle accompanied with onboard computer and sensors. The integrated mobile manipulator is used as a testbed to emulate and realize orbital motion and control algorithms. The kinematic relations of the ground vehicle, robotic manipulator and the coupled kinematics are derived. The system is used to emulate an orbit trajectory. The system is scalable and capable of emulating servicing missions, satellite rendezvous and chaser follower problems.

To my father, Sayed Abbas, and Robert Dwyer

ACKNOWLEDGMENTS

I would like to thank my adviser Dr. Tarek Elgohary for his valuable ideas, guidance and hints during the project. I appreciate all the effort and dedication by my esteemed colleagues Ryan Ketzner and Hunter Quebedeaux. I would also like to thank Dr. Das and Dr. Xu for their insights and comments during the the defense. I am thankful for my family and friends for the unbounded support during this journey. Namely, Faten Elsayed, Islam ELsadek, Ibrahim Elsadek, Deborah Carroll, Haithem Taha, Ahmed Hassan, Ahmed Aboelmakarem, Kevin Williams, Tahsinul Haque Tasif, Ryan Debevec, Nick Patel, Drew Hampton, Mohamed Ayman, and Ammar Jahin.

TABLE OF CONTENTS

LIST OF FIGURES viii

LIST OF TABLES x

CHAPTER 1: INTRODUCTION AND LITERATURE REVIEW 1

Orbit Motion Emulation 1

Omnidirectional Mobile Robots 3

Robotic Manipulators 6

Mobile Manipulators 7

CHAPTER 2: GROUND VEHICLE 10

Kinematics Model 10

Simulation and Experimental Results 15

CHAPTER 3: ROBOTIC MANIPULATOR 23

Robotic Manipulator 23

Forward Kinematics 25

Inverse Kinematics 28

Simulation Results 32

Experimental Results 36

CHAPTER 4: ROME MOBILE MANIPULATOR 40

Application to Astrodynamics 44

CHAPTER 5: CONCLUSION AND FUTURE WORK 49

LIST OF FIGURES

Figure 2.1: Kinematic arrangement of the mobile robot 11

Figure 2.2: Path following Simulink model for the ground vehicle 15

Figure 2.3: Path following controller for the ground vehicle 16

Figure 2.4: Circular trajectory kinematics simulation 16

Figure 2.5: Circular trajectory experimental results and error of the mobile robot . . 17

Figure 2.6: Ground platform CAD trimetric View 18

Figure 2.7: Simulated linear trajectory of the mobile robot 19

Figure 2.8: Linear trajectory tracking experimental results and tracking error of the Mobile Robot 20

Figure 2.9: Simulated elliptical trajectory of the mobile robot 21

Figure 2.10: Elliptical trajectory tracking experimental results and error 22

Figure 3.1: Planar two-link arm, [1] 25

Figure 3.2: AR2 wire skeleton, [2] 27

Figure 3.3: AR2 workspace 29

Figure 3.4: Resolved rate kinematic control 32

Figure 3.5: End effector position vs ramp reference using a proportional controller . 33

Figure 3.6: End effector position vs sine wave reference using a proportional controller 34

Figure 3.7: AR2 pose tracking absolute error 36

Figure 3.8: AR2 manipulator CAD, 3D printed models and electrical enclosure . . . 37

Figure 3.9: AR2 constrained motion in Z direction: ramp trajectory with open loop control 38

Figure 3.10: AR2 constrained motion in Z direction: Sinusoid experiment using PI controller 39

Figure 3.11: AR2 constrained motion in Z direction: LEO experiment using PID controller 39

Figure 4.1: ROME hardware assembly 41

Figure 4.2: ROME closed loop resolved rate block diagram 42

Figure 4.3: End effector position vs reference position using a Proportional controller 43

Figure 4.4: ROME pose tracking absolute error 43

Figure 4.5: ROME following an elliptic orbit in a decoupled fashion 46

Figure 4.6: 3D plots for ROME following an elliptic orbit 47

Figure 4.7: Error in ground plane motion produced by ground vehicle 48

LIST OF TABLES

Table 3.1: DH parameters definition, [1] 23

Table 3.2: Two-link arm DH parameters 24

CHAPTER 1: INTRODUCTION AND LITERATURE REVIEW

Orbit Motion Emulation

The design and testing of spacecraft control algorithms is a challenging task and has been the subject of research and development by aerospace engineers for decades. The availability of a platform that emulates orbital motion and tests the performance and reliability of control algorithms is favorable. Simulating orbital motion in a laboratory environment is good practice to develop and test reliable control algorithms in real-time. The process usually includes different kinds of robots to achieve a specific task planned to be carried out in space. Another approach for testing control algorithms implementation in real-time is Hardware in the Loop (HIL) simulations. The process of solving the system's equations are carried out on an embedded computer to test the performance and reliability of the hardware and the algorithms to be used in space missions. Many researchers investigated and developed various systems to emulate space vehicles and satellite motion. Choon et al. designed and fabricated two air-bearing vehicles to serve as the main part of a satellite maneuver testbed, [3]. Other kinds of simulators were developed to simulate the docking problem. For example, the European Proximity Operations Simulator (EPOS) developed by the German Aerospace Center uses two robotic manipulators each with 6-DOF fixed on a slide. EPOS allows docking and rendezvous real-time simulations, [4]. Ananthakrishnan et al. emulated on-orbit contact dynamics in a one-G ground system by modeling and testing feed-forwared filter based on prediction. The International Space Statoin (ISS) and spacecraft contact dynamics were simulated and illustrated using a 6-DOF Stewart platform, [5]. In order to simulate contact tasks carried out by the special purpose dexterous manipulator, a robotic arm on the ISS, the Canadian Space Agency developed a task verification facility utilizing a 6-DOF hydraulic robot, [6]. US Naval Research Lab uses two 6-DOF robotic arms for satellite rendezvous simulation to test rendezvous sensors, [7]. Bai et al. presented a high fidelity dynamical model for an autonomous mobile robotic system to emulate spacecraft motion. They used a mobile omnidirectional base robot with a 6-DOF Stewart platform, [8]. Fouse et al. proposed an electromagnetic docking system to mitigate the uncertainty in high risk operation of small satellites docking. The proposed system is able to dock at various orientations and is tolerant towards minor misalignments, [9].

In order to emulate the process of removing orbital debris, Papadopoulos et al. presented a hardware emulator consisting of a robot mounted on air bearings and propelled by PWM thruster forces, [10]. Their preliminary results showed potential for the system to be used reliably for emulation purposes. Cavalieri et al. presented a guidance, navigation and control product that supports a ground-based simulation necessary for removing orbital debris autonomously, [11]. They conducted two laboratory experiments and the results verified the operation of the system. Mao and Wang investigated the concept of space manipulator-based microgravity platform. They used a system that consists of a manipulator with an end effector shield and a testbed floating in it. The orbit was assumed to be circular and Clohessy-Wiltshire equations were utilized. Another case where the initial relative speed is in the opposite direction of the last link was tested. A routine is engaged to test the reachability of the relative motion by the robotic arm, [12].

In order to ensure equilibrium across the range of robot motion, gravity compensation is a widely used technique to reduce the actuator loads generated by the robot links weight. Various operations are achieved at a low motor speed which generate high gravitational torques. In such a case, gravity compensation is favorable in order to reduce the torques required to operate the robot. The nature of the forces needed to compensate gravity in a robotic system may be diverse. Mechanical gravity compensation techniques include gravity compensation by counterweights, gravity compensation by springs, and gravity compensation by using auxiliary actuators. Optimal control of input torques can also be used to cancel the gravity effects on a robotic system link, [13]. Adding counterweights to the robotic system is a classical method applied moving links to keep their center of masses stationary. Hand-operated balanced manipulators successfully utilized gravity compensation. The balanced manipulator has a simple mechanical system used for handling an object in any position of the workspace while maintaining balance. Constantly canceling gravity allows handling objects manually, [14]. Auxiliary linkages, any mechanical system linking the initial structure of the robot with a balancing member, connected with the initial system are another technique used to maintain balance. Using such mechanisms can reduce the consumption of power and energy, [15].

Omnidirectional Mobile Robots

The term mobile robot is usually interpreted as a gadget that can move self-rulingly from place to place to accomplish various objectives. Mobile robots are utilized in a wide scope of applications including production lines, military tasks, medicinal services, pursuit and save, and in homes for cleaning and lawn mowing. Automated guided vehicles or automatic guided vehicles (AGVs) were developed in 1953. AGVs are regularly utilized in modern applications to move materials around an assembling office or a stockroom. Ordinary AGV types are tuggers (AGVs that pull trucks), unit loaders (AGVs with installed roller tables for parts-plate exchanges), and fork trucks (robots like manual fork trucks). The utilization of portable robots, and AGVs specifically, is developing as the scope of robot applications in processing plants, medical clinics, places of business, and so forth widens. Portable robots can utilize a scope of movement systems, for example, flying, swimming, slithering, and strolling, [16].

The design and development of ground vehicles and their wheels have been under constant progression in the past decades. For manmade vehicles and mobile robotics, wheels have been the number one choice for designers and engineers to achieve motion with good efficiency and simple mechanism implementation. Unlike legged robots, for instance, the stability study is not of an issue for wheeled robots because all the wheels touch the ground. Hence, three wheels are sufficient to guarantee a stable equilibrium of the vehicle, [17].

Four wheel types are generally used for ground robots; Castor wheel, Swedish (Omnidirectional) wheel, and ball wheel. The standard wheel has 2-DOF and rotates around the axle. The Castor wheel has 2-DOF and rotates around a steering joint with an offset. The Swedish wheel has 3-DOF rotating around the axle, rollers and contact point. The ball wheel is basically a ball that supports the system and rolls in all in-plane directions. It is technically difficult to be realized.

The Swedish wheel and the ball wheel are less constrained in motion that the other wheel types. The Swedish wheel acts as a normal wheel while providing a low resistance in a different direction. This direction can be 90 degrees or 45 degrees depending on the type of the Swedish wheel. The only actively powered joint is around the primary axis of the wheel and the small rollers around the circumference are passive. The main advantage of Swedish wheel is that the wheel is able to move kinematically with a very low friction along various possible trajectories while being powered only from one axis. The choice of wheels depends on the choice of the wheel arrangement that fits the mission requirements. Maneuverability, controllability and stability are the main characteristics for the process of choosing a robot that fits the mission. Conventionally, three wheels are essential for static stability and the center of gravity has to lie the triangle formed by the wheels. For maneuverability, omnidirectional robots can move in the (x, y) plane in any direction regardless of the robot orientation around its vertical axis. Omnidirectional robots usually utilize omnidirectional wheels or ball wheels in order to achieve the 3-DOF motion. On the other hand, castor wheels can be used in a four wheel arrangement to achieve the same kind of motion. One of the prominent examples of using castor wheels for space motion emulation is the ones designed at Texas A&M, [8]. Moreover, a power castor wheel developed by Holmberg and Khatib was used in the XR4000 mobile robot, [18]. The controllability of the four castor wheel arrangement is lower. Generally speaking, the relation between controllability and maneuverability is inversely proportional. The process of translating required translational and rotational velocities to commands in a four castor wheel arrangement is more significant, [17].

The Cye personal robot is a nonholonomically constrained differential drive running on two wheels. It is designed for home, office or research use. Main wheels provide balance for the robot and its only mean of navigation is dead reckoning which is calculating the vehicle's current position by utilizing the history of position and propagating that position based on estimated speeds along time and course. Dead reckoning is advantageous when it comes to computation simplicity. However, the main disadvantage with dead reckoning is the accumulation of error with time. Batavia and Nourbakhsh developed a path planning algorithm for the Cye robot. Their optimization criterion allowed the robot to find short paths while avoiding obstacles with suitable distance. They demonstrated the robustness of their planner by conducting 200 real world experiments, [19].

In order to move an omnidirectional robot in a specific direction, more complex control algorithms are required for the less maneuverable robots. The Tribolo designed at the Swiss Federal Institute of Technology is based on three independently actuated wheels. This arrangement provides excellent maneuverability but its limitations include the inability to run on irregular surfaces or carrying big loads.

Uranus is another robot built by Carnegie Mellon with a four Swedish wheel arrangement. It is composed of four independently motored 45-degree Swedish wheels. The robot can be moved along any trajectory while spinning around its vertical axis by utilizing the direction and relative speeds of the four wheels. This design makes the robot an over-actuated system which is not minimal in terms of control, [20].

Williams et al. presented a dynamical model for holonomic wheeled robots with wheel slip. They validated their model by experimentally measuring friction coefficients and quantifying the force to prompt slip. Simulations showed that the initial models disagreed with the experimental results. The results from this model and the experiment agreed, [21]. Watanabe et al. derived the dynamics of an omnidirectional robot with lateral orthogonal-wheels. The resolved acceleration control system was derived and showed good efficiency, [22].

Rajagopalan developed kinematic expressions of wheeled mobile robots different scenarios for steering and driving, [24]. Shekar derived a wheel slip dynamic model using accessibility and controllability in nonlinear control theory, [25]. Balakrishna and Ghosal modeled traction taking slip into consideration for nonholonomic mobile robots, [26]. Scheding et al presented experimental validation of a navigation system system that can handle autonomous vehicle wheel slip by using multi-sensor feedback, [27].

Generally, there are some standard controllers that have been designed and implemented like proportional plus integral controllers, Lyapunov based controllers, resolved motion rate controllers and resolved motion acceleration controllers. The controllers mentioned involve two stages. The first is kinematic tracking which uses the kinematic model of the robot and used a suitable Lyapunov function to control the linear and angular velocities. The second uses Lagrange model which includes the motor dynamics and the gearbox, [28].

Robotic Manipulators

Robot manipulators have various applications like assembly jobs in factories, welding and handling tasks, and recently operating surgeries in operation rooms. A very common arrangement of robotic manipulators is the 6 degree of freedom robot that consists of connected rigid links actuate by motors fixed at joints. Robotic manipulators are fixed in position for specific tasks which limits its workspace. The Selective Compliance Assembly Robot Arm (SCARA) is rigid vertically but is compliant in the horizontal position which suits tasks carried out in-plane such as electronic boards manufacturing. A large working volume can be achieved by confining the motion of a robotic arm to one or 2-DOF along overhead rails; this kind of robots is usually called a gantry robot. A parallel-link manipulator has links connected to the end effector in parallel; this arrangement is advantageous in having the motors fixed in the base providing a rigorous structure.

Safety and perception are simplified by having the robot immobile. Factory robots can be fast and precise because of the orderly, well set work environment and the assumption that the location of objects is well known. These objects are manipulated by a robot's end effector. The end effector is usually the tool that will carry out the task. The problem of robotic manipulation is till undergoing research and development for better performance and newer tasks.

Ruehl et al. presented a test bed for robotic manipulation in microgravity to simulate on-orbit-servicing missions, [29]. Their platform uses the KUKA-DLR which is an iteration of the current KUKA LBR iiwa lightweight robot. Boasting precision within 0.1 millimeters, this choice is highly precise. However, the KUKA LBR suite of manipulators range around $200,000: well outside the feasible range ROME aims to achieve. Peterson et al proposed a robust platform in which a hybrid dynamical system is used to switch between various algorithms. The manipulator hardware used for this application was a PUMA 560 arm. Being another industrial arm, the PUMA arm has repeatability within 0.1 millimeters, [30].

Mobile Manipulators

Mobile manipulators are robotic manipulators fixed on top of moving bases. They are widely used in agriculture, space applications and warehouses for big retail companies. Hamner et al developed a mobile manipulator for insertion assembly missions. Using reactive task control strategies allowed the system to overcome inherent uncertainties and exceptions, [31]. Peterson et al. and Nagatani et al developed a mobile manipulator utilizing force and torque control to open a door, [30, 32]. Chitta et al. used a graph-based representation to plan arm and base motion to open a variety of door types, [33]. Kinematic redundancy resolution of mobile manipulators for material handling was studied by Pin and Culioli, [34]. In order to optimize the the configuration and position of the system during movement, they studied its utility to take care of changes in task constraints or requirements. Obstacle avoidance and maneuverability functions were investigated. Adavit and Kemp developed a mobile manipulator and called it EL-E. The robot can approach the location of an object illuminated by a laser pointer and grasp it, [35]. Agah and Tanie presented control problems accompanied with a mobile manipulator giving objects to Humans, [36]. Tomizawa et al. described a mobile manipulator located in a library to help humans browse books, [37]. Borth and Pobil demonstrated a voice guided mobile manipulator that can verbally interact with the user to identify and grasp objects, [38]. Holmbery and Khatib designed a powered caster vhiecle (PCV) demonstrating smooth and accurate motion. The system can be controlled. Moreover, they proposed a novel approach to model and control parallel redundant system dynamics, [18]. Khatib et al developed models and methodologies for analyzing and controlling new multiple mobile manipulation systems. The four fixed base manipulation methods were extended to Stanford Assistant Mobile Manipulators (SAMM). Osumi et al studied cooperating control of multiple mobile manipulators, [39]. Two mobile base robots performed assembly tasks following a behavior-based system, [40].

Several studies discussed the problem of configuration optimization. Yamamoto and Yun developed an algorithm to plan and control the manipulator to be positioned at the preferred configurations measured by its manipulability. The simulation results proved the desired algorithm result. Consequently, the algorithm was verified in real time on a mobile manipulator system, [41]. A multi-criteria motion planning of a mobile manipulator systems was carried out by using genetic algorithms. The multi-criteria were based on manipulability, distribution of torque and obstacle avoidance, [42].

The availability of a platform that emulates orbital motion and tests the performance and reliability of control algorithms is favorable. Emulating orbital motion in a laboratory environment is an important process to develop and test reliable control algorithms and sensors in real time. The benefit of hardware emulation over software simulation is the ability for engineers to rapidly control prototype and iterate their control strategies on real-world hardware interfaces. One of the main challenges in building such platforms is the high cost and the high maintenance expenses. In this work, we present a mobile manipulator that consists of a 6-DOF robotic manipulator fixed on top of a holonomic ground vehicle. The 9-DOF Rapid Orbital Emulator (ROME) is capable of simulating orbital motion in 3 dimensional space. The forward and inverse kinematics of the two robots as well as the mobile manipulator are derived. Real time experiments for closed loop control on the ground vehicle, robotic manipulator and mobile manipulator were carried out. The system integration showed stability with minor vibrations. The trajectory generated for the experiment is the solution of an elliptic orbit. The solution was carried out using the classical Lagrange Gibbs method.

The thesis is divided into four main chapters. Chapter two discusses the forward and inverse kinematics, simulation and experimental results of the ground vehicle. The robotic manipulator forward kinematics, path following simulations and experiments are demonstrated in chapter 3. Chapter 4 presents the mobile manipulator coupled forward kinematics, path following simulations, experiments and applications to astrodynamics. Finally, concluding remarks and future work are presented in chapter 5.

CHAPTER 2: GROUND VEHICLE

Kinematics Model

The forward kinematics of a three wheeled mobile robot can be described as

$$\boldsymbol{\chi} = \boldsymbol{f}(\boldsymbol{q}) \tag{2.1}$$

where the joint space $\boldsymbol{q}$ is the set of three motors and wheels, and the Cartesian space $\boldsymbol{\chi}$ is the general 6-DOF vector. An easy and intuitive way of describing this relation is by defining poses, the combination of position and orientation, in terms of homogeneous transformation matrices.

$$\boldsymbol{T}_{GV}^{I} = \begin{bmatrix} \boldsymbol{R}_{GV}^{I} & \boldsymbol{o}_{GV}^{I} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2.2}$$

where $\boldsymbol{R}_{GV}^{I}$ is the 3×3 rotation matrix and $\boldsymbol{o}_{GV}^{I}$ is the position of the origin of the frame of reference $\{GV\}$ with respect to the origin of the frame of reference $\{I\}$. The model setup is based on an inertial frame of reference $\{I\}$ which includes two axes (X_I, Y_I) and the ground vehicle frame of reference $\{GV\}$. The mobile robot body frame is located at the center of gravity of the vehicle with the X axis passing through wheel number 1 and the Y axis perpendicular to it and located between wheel number 1 and wheel number 2 as shown in Fig. 2.1, [43]. The rotation from the ground vehicle frame to the inertial frame can be described by the rotation matrix

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \tag{2.3}$$

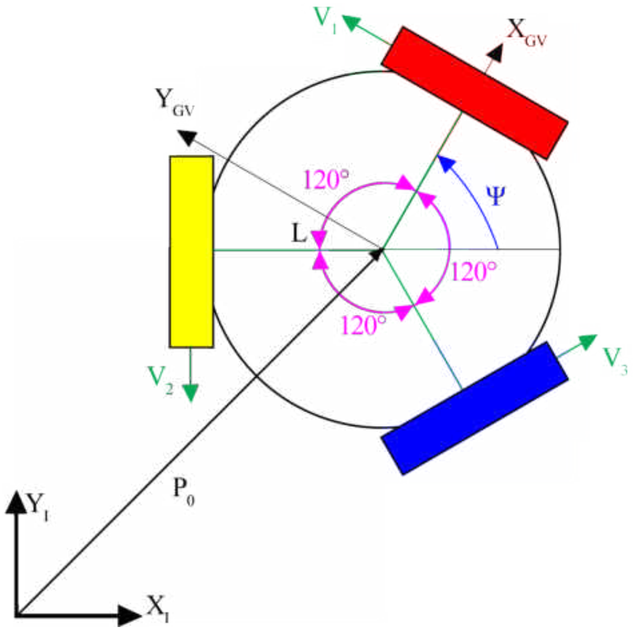


Figure 2.1: Kinematic arrangement of the mobile robot

where ψ is the rotation angle measured from the horizontal axis X and positive in the counterclockwise direction. The position of the center of mass of the vehicle can be described in the inertial frame as

$$\boldsymbol{p}_{GV}^{I} = xX_I + yY_I \tag{2.4}$$

The homogeneous transformation matrix that takes into account the rotation and translation of the ground vehicle frame with respect to the inertial frame can be written as

$$\boldsymbol{T}_{GV}^{I} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & x \\ \sin\psi & \cos\psi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

Consequently, the position of each wheel can be described in terms of its angular location and distance from the origin of the center of mass of the ground vehicle as

$$\boldsymbol{p}_i^{GV} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \boldsymbol{R}(\zeta) L \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.6}$$

where L is the distance from the center of mass to each wheel and ζ is the constant angular location of each wheel measured counterclockwise from the inertial horizontal axis X_I. For the three wheels

$$\begin{aligned} \boldsymbol{p}_1^{GV} &= L\cos(0) X_{GV} + L\sin(0) Y_{GV} \\ \boldsymbol{p}_2^{GV} &= L\cos\left(\frac{2\pi}{3}\right) X_{GV} + L\sin\left(\frac{2\pi}{3}\right) Y_{GV} \\ \boldsymbol{p}_3^{GV} &= L\cos\left(\frac{4\pi}{3}\right) X_{GV} + L\sin\left(\frac{4\pi}{3}\right) Y_{GV} \end{aligned} \tag{2.7}$$

The translational direction unit vector of each wheel with respect to the ground vehicle frame can be described by the vector

$$\boldsymbol{d}_i^{GV} = \frac{1}{L}\boldsymbol{R}\left(\frac{\pi}{2}\right) p_i^{GV}, \quad i = 1, 2, 3 \tag{2.8}$$

The position and velocity of each wheel with respect to the inertial frame can now be expressed as

$$\boldsymbol{r}_i^I = \boldsymbol{p}_{GV}^I + \boldsymbol{R}\left(\psi + \frac{2\pi}{3}(i-1)\right)\boldsymbol{p}_i^{GV} \tag{2.9}$$

$$\boldsymbol{v}_i^I = \dot{\boldsymbol{p}}_{GV}^I + \dot{\boldsymbol{R}}\left(\psi + \frac{2\pi}{3}(i-1)\right)\boldsymbol{p}_i^{GV} \tag{2.10}$$

The translational velocity of each wheel can be described as follows

$$\begin{aligned}\boldsymbol{V}_i &= \boldsymbol{v}_i^{I^T}\boldsymbol{R}\left(\psi + \frac{2\pi}{3}(i-1)\right)\boldsymbol{d}_i \\ &= \dot{\boldsymbol{p}}_{GV}^{I^T}\boldsymbol{R}\left(\psi + \frac{2\pi}{3}(i-1)\right)\boldsymbol{d}_i + \boldsymbol{p}_i^{GV^T}\dot{\boldsymbol{R}}^T\left(\psi + \frac{2\pi}{3}(i-1)\right)\boldsymbol{R}\left(\psi + \frac{2\pi}{3}(i-1)\right)\boldsymbol{d}_i\end{aligned} \tag{2.11}$$

The translational velocities of the three wheels can then be written as

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} R\omega_1 \\ R\omega_2 \\ R\omega_3 \end{bmatrix} = \boldsymbol{P}(\psi)\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} \tag{2.12}$$

where R is the wheel radius and ω_i is the angular velocity of the three wheels and

$$\boldsymbol{P}(\psi) = \begin{bmatrix} -\sin\psi & \cos\psi & L \\ -\sin(\frac{\pi}{3}-\psi) & -\cos(\frac{\pi}{3}-\psi) & L \\ \sin(\frac{\pi}{3}+\psi) & -\cos(\frac{\pi}{3}+\psi) & L \end{bmatrix} \tag{2.13}$$

The matrix $\boldsymbol{P}(\psi)$ is always nonsingular for any ψ and its inverse looks as follows

$$\boldsymbol{P}^{-1}(\psi) = \begin{bmatrix} -\frac{2}{3}\sin\psi & -\frac{2}{3}\sin(\frac{\pi}{3}-\psi) & \frac{2}{3}\sin(\frac{\pi}{3}+\psi) \\ \frac{2}{3}\cos\psi & -\frac{2}{3}\cos(\frac{\pi}{3}-\psi) & -\frac{2}{3}\cos(\frac{\pi}{3}+\psi) \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \tag{2.14}$$

This model allows for trajectory tracking control along an arbitrary time-varying path according to the control law

$$\boldsymbol{V} = \boldsymbol{P}(\psi)\left(-\boldsymbol{K}_p\boldsymbol{\chi}_e - \boldsymbol{K}_i\int\boldsymbol{\chi}_e dt + \dot{\boldsymbol{\chi}}_d\right) \tag{2.15}$$

where $\boldsymbol{\chi}$ is the states vector, $\boldsymbol{\chi}_e$ is the error vector and $\dot{\boldsymbol{\chi}}_d$ is the desired velocity vector

$$\begin{bmatrix} x_e(t) \\ y_e(t) \\ \psi_e(t) \end{bmatrix} = \begin{bmatrix} x(t) \\ y(t) \\ \psi(t) \end{bmatrix} - \begin{bmatrix} x_d(t) \\ y_d(t) \\ \psi_d(t) \end{bmatrix} \tag{2.16}$$

and the motor output voltage $\boldsymbol{u}(t)$ can be calculated using the control law

$$\boldsymbol{u}(t) = \boldsymbol{K}_p\boldsymbol{\chi}_e(t) + \boldsymbol{K}_i\int\boldsymbol{\chi}_e(t)dt \tag{2.17}$$

In order to prove the stability of the controller used, consider the Lyapunov function

$$\mathcal{V} = \frac{1}{2}\boldsymbol{\chi}'\boldsymbol{\chi} + \frac{1}{2}\int(\boldsymbol{\chi}_e')dt\boldsymbol{K}_I\boldsymbol{\chi}_e \tag{2.18}$$

$$\begin{aligned} \dot{\mathcal{V}} &= \boldsymbol{\chi}'\dot{\boldsymbol{\chi}} + \int(\boldsymbol{\chi}_e)dt\,\boldsymbol{K}_I\boldsymbol{\chi}_e \\ &= \boldsymbol{\chi}'\left(-\boldsymbol{K}_p\boldsymbol{\chi}_e - \boldsymbol{K}_I\int(\boldsymbol{\chi}_e)dt\right) + \boldsymbol{\chi}_e'\boldsymbol{K}_I\boldsymbol{\chi}_e) \\ \dot{\mathcal{V}} &= -\boldsymbol{\chi}'\boldsymbol{K}_P\boldsymbol{\chi} < 0 \end{aligned} \tag{2.19}$$

The controller drives the system to a stable equilibrium for $\boldsymbol{K}_p$ is a positive definite matrix.

Simulation and Experimental Results

The kinematics were modeled in Simulink and used to produce linear, circular, and elliptical trajectories as shown in Fig. 2.2. The path following controller used for the ground vehicle is a proportional integral controller as shown in Fig. 2.3.

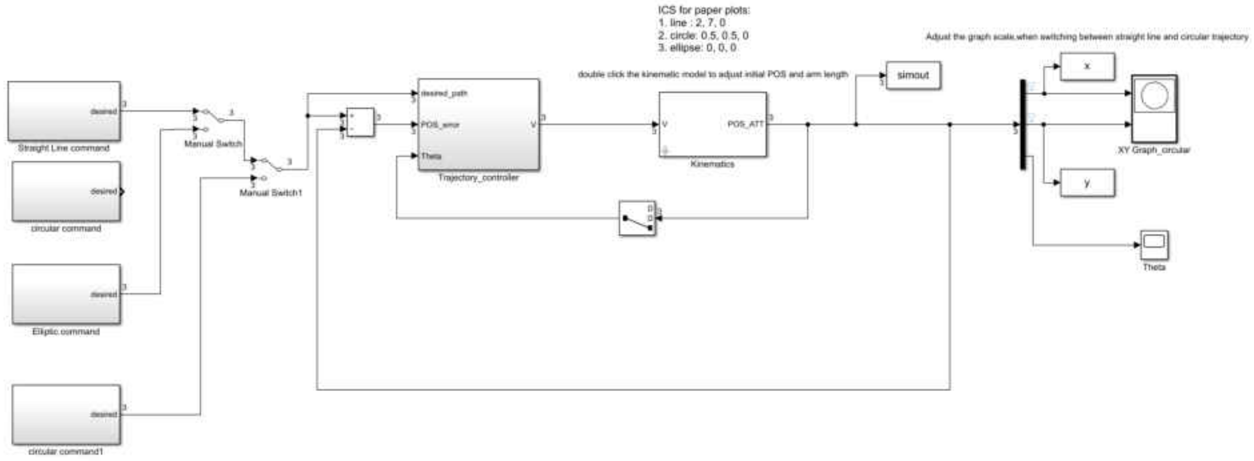


Figure 2.2: Path following Simulink model for the ground vehicle

These trajectories were reproduced using the ROME ground platform. For the circle, the trajectory is given as

$$x(t) = cos(.25t) \qquad y(t) = sin(.25t) \qquad \psi(t) = 0 \tag{2.20}$$

The model follows the circular path smoothly as shown in Fig. 2.4. The absolute error plots are generated by subtracting the state value from the reference trajectory value.

The ROME ground vehicle uses three omnidirectional wheels in order to allow holonomic motion in the XY plane. The platform consists of a modified Nexus Robot Kit 10013, using an Arduino

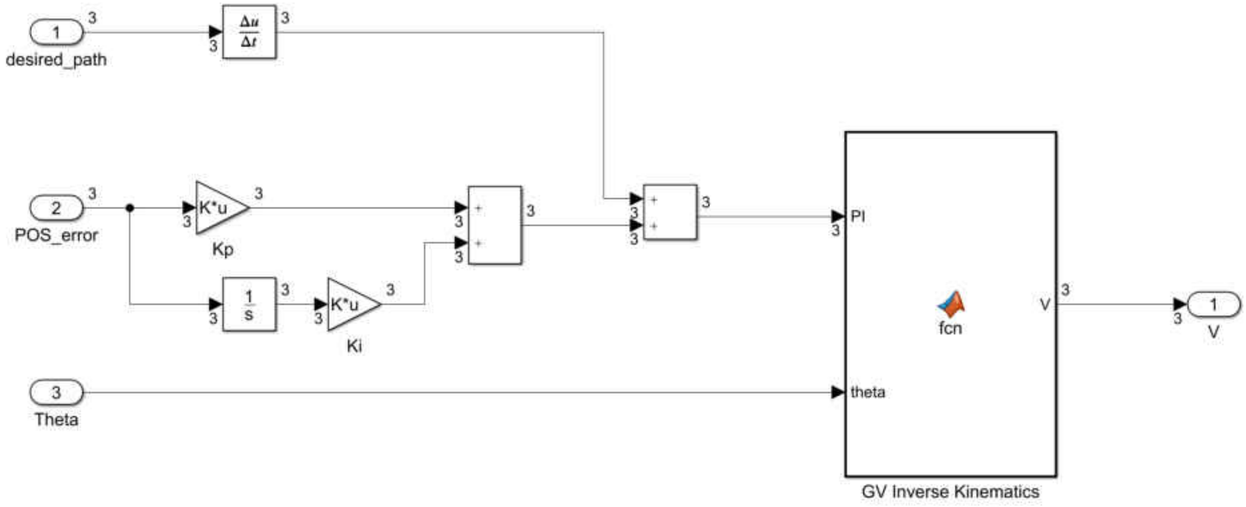
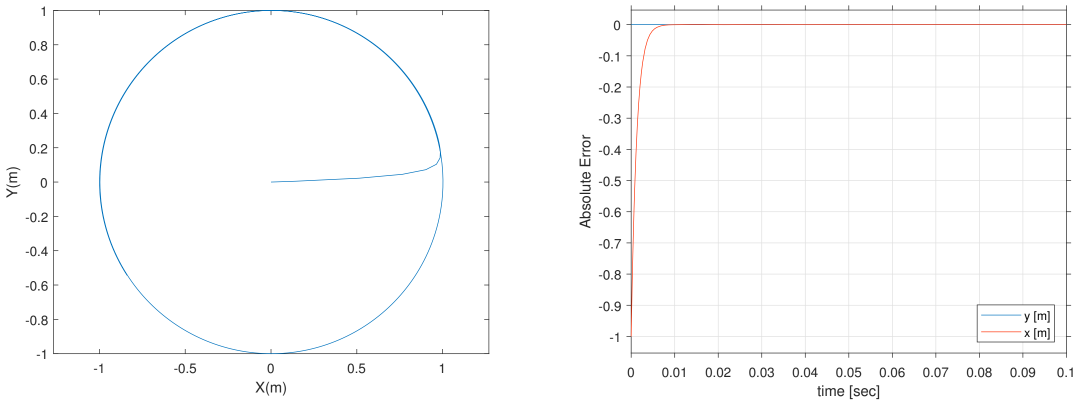


Figure 2.3: Path following controller for the ground vehicle

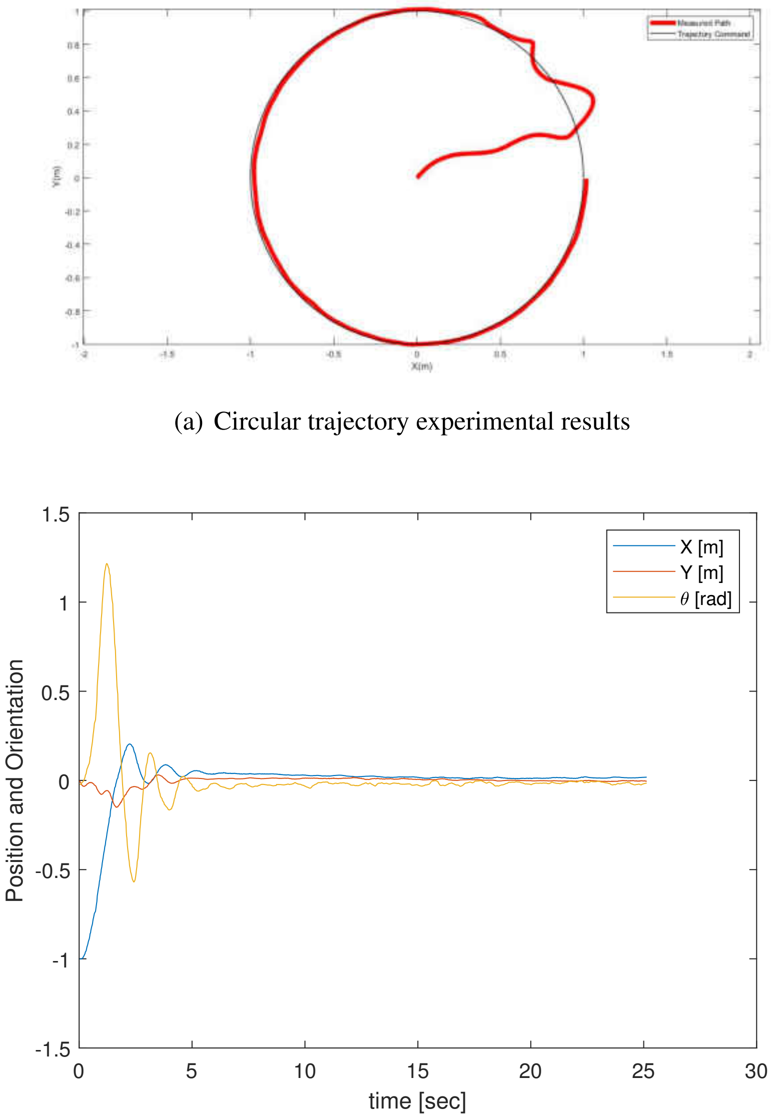


(a) circular trajectory

(b) Absolute error vs. time

Figure 2.4: Circular trajectory kinematics simulation

MEGA and an Adafruit V2 Motor Shield to control PWM voltages to the three motors in place of the stock controller and driver. Holes were drilled to the top of the platform to allow mounting the robotic arm. The ground vehicle uses optical encoders on the motors for velocity feedback and



(a) Circular trajectory experimental results

(b) Circular trajectory experimental error

Figure 2.5: Circular trajectory experimental results and error of the mobile robot

an Optitrack Prime 13W camera system for position feedback. Motive, Optitrack's accompanying software, allows position and orientation data to be streamed to Matlab in real time at 240 frames per second. Motor velocity control is handled by the Arduino, which receives new setpoints over serial through Matlab using the Matlab Arduino support package.

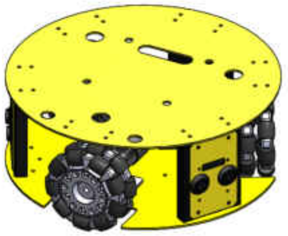


Figure 2.6: Ground platform CAD trimetric View

The experiment showed that the mobile robot can follow a circular path with some transient error at the beginning of the maneuver as presented in Fig. 2.5.

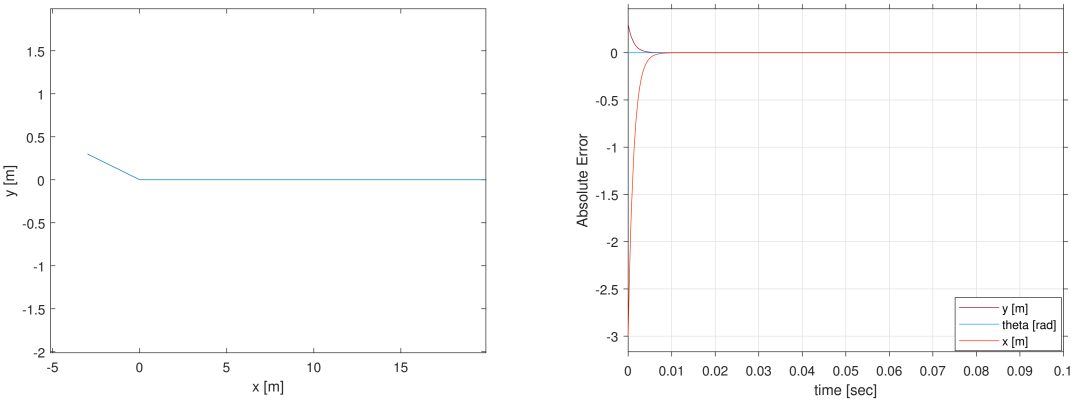
For the line, the trajectory is given as

$$x(t) = t \qquad y(t) = 0 \qquad \psi(t) = 0 \tag{2.21}$$

The results from the line following simulation and experiment agree to as shown in Fig. 2.7 and 2.8. The line following maneuver remains as one of the simplest maneuvers for the ground platform to achieve. For the ellipse, the trajectory is given as

$$x(t) = cos(0.25t) \qquad y(t) = 1.5sin(0.25t) \qquad \psi(t) = 0 \tag{2.22}$$

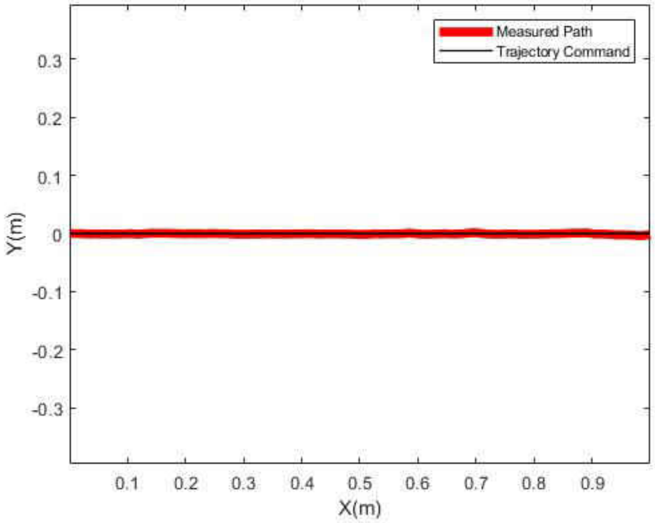
Both the simulation and experiment of the elliptic trajectory agree as shown in Fig. 2.9 and 2. The mobile robot kinematic model showed good accuracy compared to the experimental realization of the prescribed maneuvers.



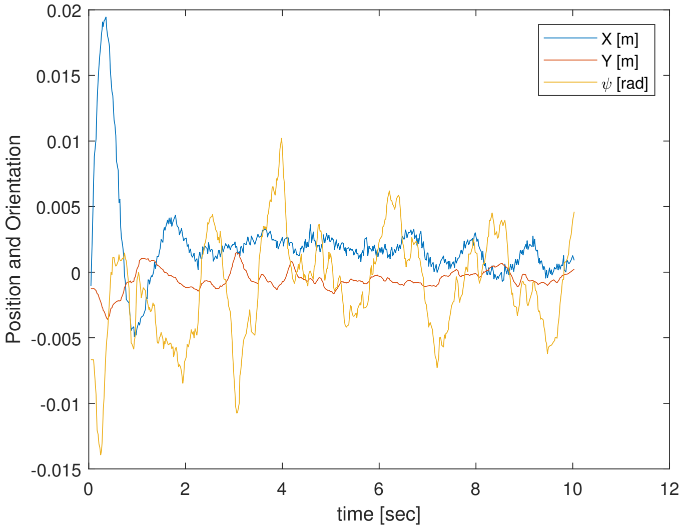
(a) linear trajectory

(b) Absolute error vs. time

Figure 2.7: Simulated linear trajectory of the mobile robot

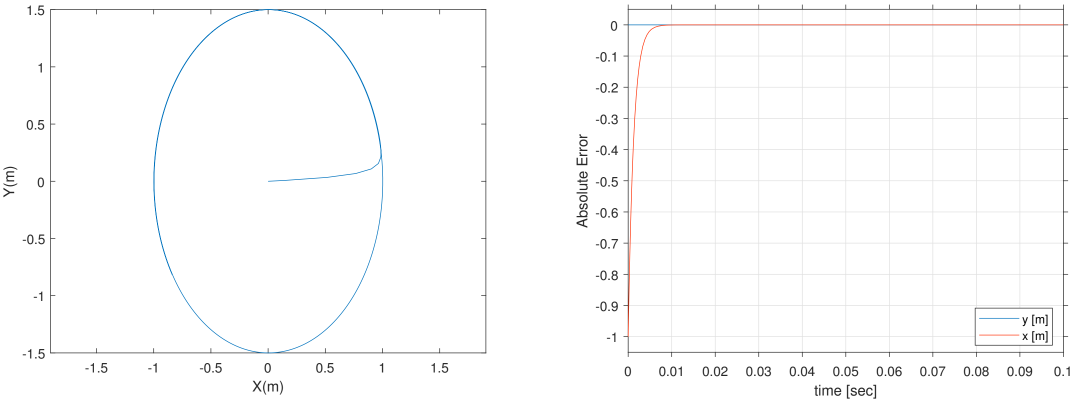


(a) Straight line tracking experiments results



(b) Straight line tracking experimental error

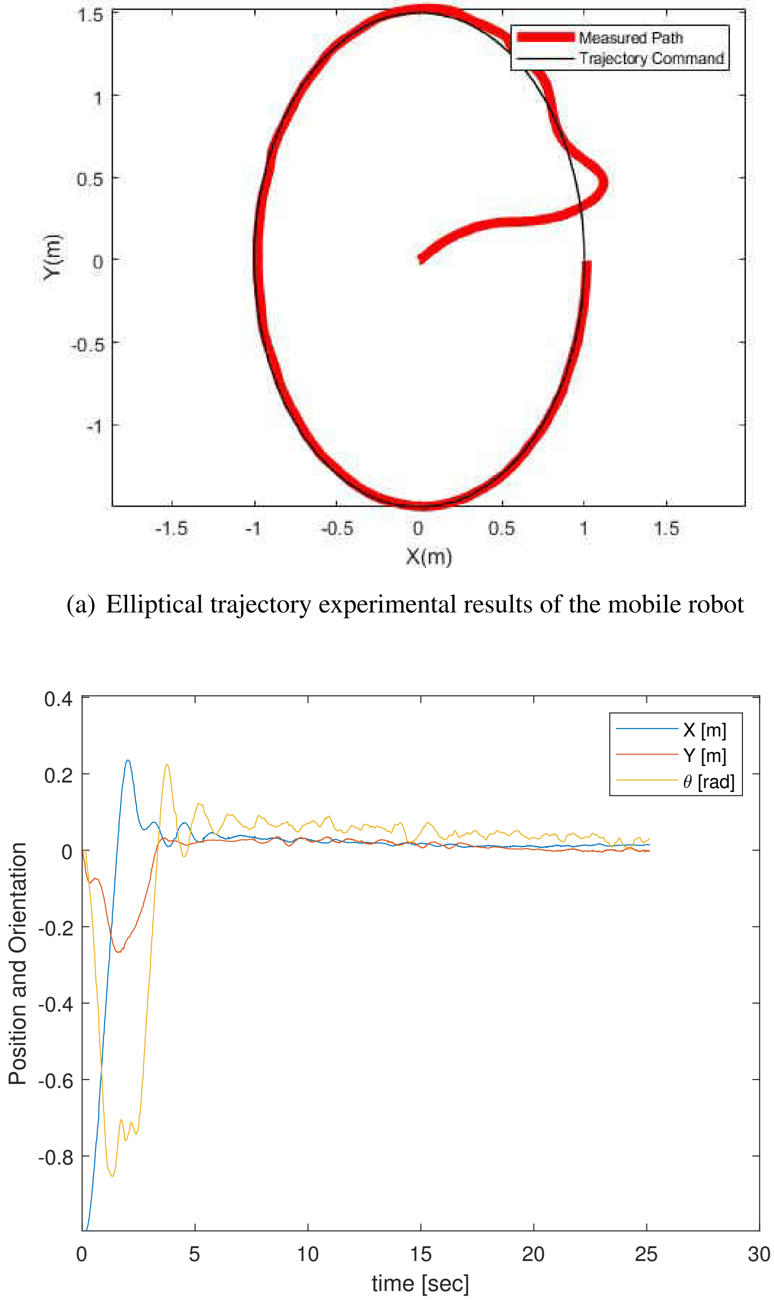
Figure 2.8: Linear trajectory tracking experimental results and tracking error of the Mobile Robot



(a) Elliptic trajectory

(b) Absolute error vs. time

Figure 2.9: Simulated elliptical trajectory of the mobile robot



(a) Elliptical trajectory experimental results of the mobile robot

(b) Error in elliptical trajectory tracking experiment

Figure 2.10: Elliptical trajectory tracking experimental results and error

CHAPTER 3: ROBOTIC MANIPULATOR

Robotic Manipulator

The serial chain of links are conveniently described using the Denavit-Hartenberg (DH) notation which is based on a unique setup of the frames of references along the joints in order to reduce the amount of computation needed to solve the kinematics. The DH notation can be defined as shown in Table 3.1

DH parameters can be used to construct homogeneous transformation matrices that can be used to concisely derive the kinematics of a serial chain manipulator. The homogeneous transformation matrix can be created based on a unique sequence of motions and rotations along the serial link chain. The sequence is a rotation around z, then a translation along the z axis, followed by a

Table 3.1: DH parameters definition, [1]

Joint angle	θ_j	the angle between the $x_j - 1$ and x_j axes about the z_{j-1} axis	revolute joint variable
Link offset	d_j	the distance from the origin of frame $j-1$ to the x_j axis along the $z_{j-1} axis$	prismatic joint variable
Link length	a_j	the distance between the z_{j-1} and z_j axes along the x_j axis; for intersection axes is parallel to $\hat{z}_{j-1} \times \hat{z}_j$	constant
Link twist	α_j	the angle from the z_{j-1} to the z_j axis about the x_j axis	constant
Joint type	σ_j	$\sigma = R$ for a revolute joint, $\sigma = P$ for a prismatic joint	constant

translation in the x direction and finally a rotation around the x axis.

$$
\begin{aligned}
\boldsymbol{DH}_i &= Rot_z(\theta_i)Trans_z(d_i)Trans_x(a_i)R_x(\alpha_i) \\
&= \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta_i & -\sin\theta_i\sin\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.1}
$$

To clarify the concept of DH parameters, a planar two-link arm is considered. The DH parameters can be constructed from the geometry of the arm shown in Fig. 3.1

The DH parameters of the arm are

Table 3.2: Two-link arm DH parameters

θ_j	d_j	a_j	α_j
q_1	0	a_1	0
q_2	0	a_2	0

The DH parameters stated in Table 3.2 can be used to construct two homogeneous transformation matrices identical to the matrix generated in Eq. (3.1). The homogeneous transformation matrix is used to describe the rotation and translation of each frame of reference with respect to the previous frame of reference. The first 3×3 submatrix is responsible for the rotation and the first 3 elements

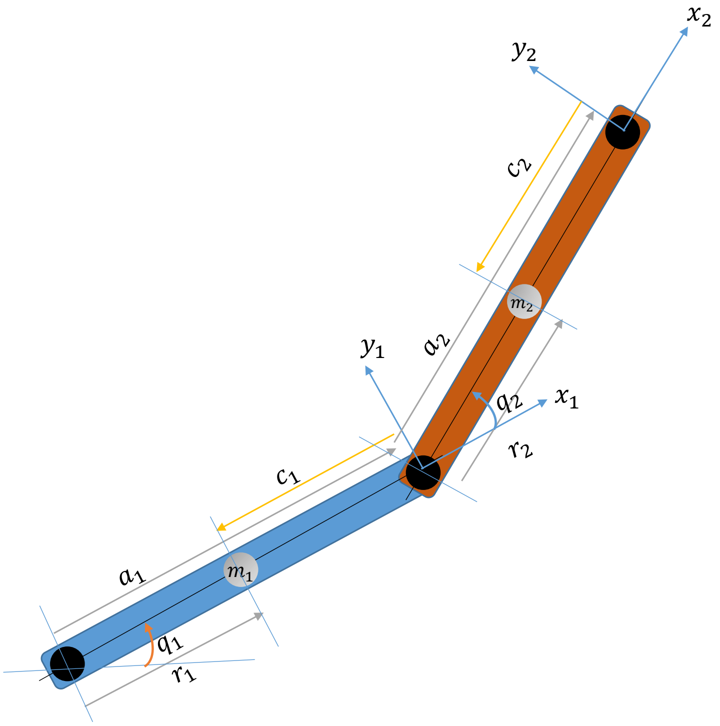


Figure 3.1: Planar two-link arm, [1]

in the $4th$ column is responsible for the translation part.

Forward Kinematics

The joint angles change according to the motors input voltage to achieve a certain task. The number of generalized coordinates is equal to the number of joints which belongs to the space of all possible configurations (a.k.a the configuration space). The pose of a manipulator in the joint space changes the position and orientation of the end effector in the task space. The task space is defined as all the possible end-effector poses. The dimension of the configuration space must be larger than or equal to the dimension of the task space. The forward kinematics describe the change in the pose of the end effector in the task space with respect to the change of the joint angles in the joint space.Hence, finding the relation between the two spaces solves the forward kinematics problem. A general form of the forward kinematic relation between the task space and joint space is

$$\boldsymbol{x} = Y(\boldsymbol{q}) \tag{3.2}$$

where $\boldsymbol{x}$ is the task variables vector and $\boldsymbol{q}$ is the joint space variables vector. For articulated robots, the serial chain of links can be conveniently described using the DH notation which is based on a unique setup of the frames of references along the joints in order to reduce the amount of computation needed to solve the kinematics. The wire skeleton with the frames of references is illustrated in Fig. 3.2. The problem of finding the position of the end effector as a function of the joint angles with respect to the base frame, defined at joint 1, is achieved by simply multiplying the six transformation matrices.

$$\boldsymbol{T}_E^B = \prod_{i=1}^{6} \boldsymbol{DH}_i \tag{3.3}$$

For each joint i, a Denavit-Hartenberg parameter matrix is created for the transformations. At joint one, frame one is known as the base frame, and at joint six, frame six is known as the tool frame. Once each frame is created, matrix products are taken down the kinematic chain of each frame. For example for the two-link manipulator the homogeneous transformation matrix is written as

$$\begin{aligned} T_2^0 &= T_1^0 T_2^1 \\ &= \begin{bmatrix} c(q_1+q_2) & -s(q_1+q_2) & 0 & a_1cq_1 + a_2c(q_1+q_2) \\ s(q_1+q_2) & c(q_1+q_2) & 0 & a_1sq_1 + a_2s(q_1+q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.4}$$

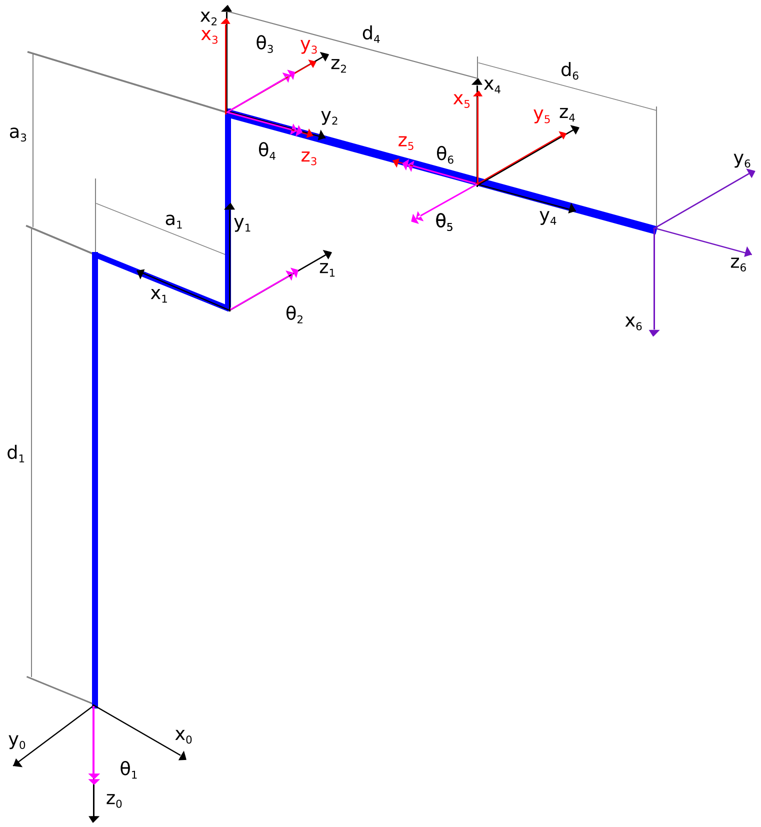


Figure 3.2: AR2 wire skeleton, [2]

Let T in Eq. (3.5) define a four by four matrix which is the product of the kinematic chain of the six total four by four Denavit-Hartenberg parameter matrices. Extracting the x, y, z, yaw, pitch, and roll data is done by examining the T matrix.

$$T = \begin{bmatrix} T_{11} & T_{12} & T_{13} & x \\ T_{21} & T_{22} & T_{23} & y \\ T_{31} & T_{32} & T_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

An interior three by three rotational matrix is embedded within the T matrix, which is used to find the yaw, pitch, and roll of the end effector. The x, y, and z positions can be observed within the T matrix. In order to extract the Eulerian angles from that matrix, a series of equations are used.

$$\begin{aligned} RotX(\alpha) &= \alpha = \text{arctan2}\left(T_{21}, T_{21}\right) \\ RotY(\phi) &= \phi = \text{arctan2}\left(-T_{31}, \sqrt{T_{32}^2 + T_{33}^2}\right) \\ RotZ(\theta) &= \theta = \text{arctan2}\left(T_{32}, T_{33}\right) \end{aligned} \tag{3.6}$$

The forward kinematics relations can be utilized to visualize the robot's workspace. The workspace of a manipulator is the total volume the end effector can reach. The manipulator geometry and the joints constraints limits the the workspace, [44]. The workspace of the AR2 robot is shown in Fig. 3.3 The workspace plot was generated by solving the inverse kinematics for configurations generated using a random number generator and bounded by the limits of each joint. The white portion in the plot determines that this volume of space is unreachable by the robot due to its design and joint limits. The dense areas in the center are more points plotted by the solver but the whole volume including the less dense points and the high density points is reachable.

Inverse Kinematics

The inverse kinematics problem can be stated based on the definition used in Eq. (3.2) as follows

$$\boldsymbol{q} = \boldsymbol{g}(\boldsymbol{x}) \tag{3.7}$$

Having a closed form system of equations describing the joint space variables in terms of the task space variables is very difficult. An effective way of solving the inverse kinematics problem is

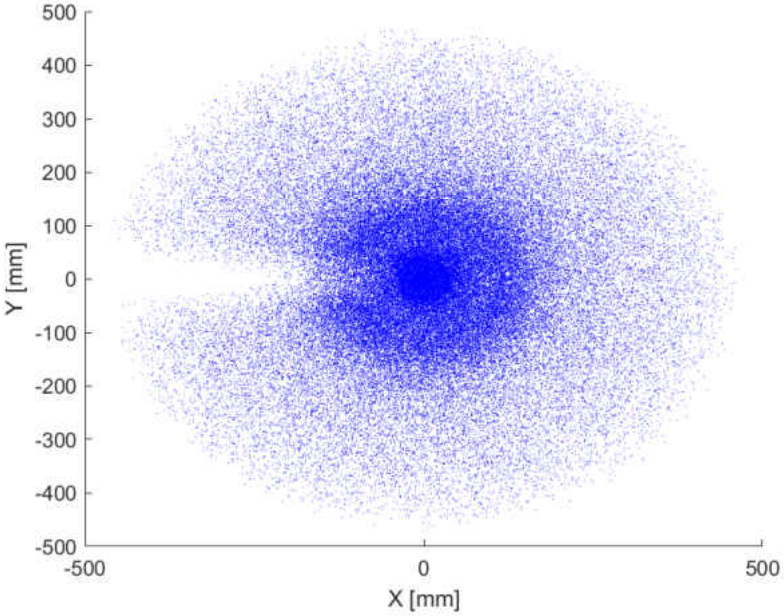


Figure 3.3: AR2 workspace

by introducing the Jacobian. The Jacobian is defined as the matrix that maps the change in the configuration space to the task space. From Eq. (3.2)

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(q)\dot{\boldsymbol{q}} \tag{3.8}$$

where the $\boldsymbol{J}$ is the Jacobian matrix and can be defined as

$$\boldsymbol{J}(q) = \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{q}} \tag{3.9}$$

To elaborate on how to calculate the Jacobian matrix analytically, consider the two-link manipulator example. From the homogeneous transformation matrix (3.4), the position of the end effector is written as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 \cos q_1 + a_2 \cos(q_1 + q_2) \\ a_1 \sin q_1 + a_2 \sin(q_1 + q_2) \end{bmatrix} \tag{3.10}$$

by differentiation

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -a_1 \dot{q_1} \sin q_1 - a_2 (\dot{q_1}) + \dot{q_2}) \sin(q_1 + q_2) \\ a_1 \dot{q_1} \cos q_1 + a_2 (\dot{q_1}) + \dot{q_2}) \cos(q_1 + q_2) \end{bmatrix} \tag{3.11}$$

The last equation is rewritten as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -a_1 \sin q_1 - a_2 \sin(q_1 + q_2) & -a_2 \sin(q_1 + q_2) \\ a_1 \cos q_1 + a_2 \cos(q_1 + q_2) & a_2 \cos(1_1 + q_2) \end{bmatrix} \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \end{bmatrix} \tag{3.12}$$

The Jacobian matrix of the two-link manipulator is

$$\boldsymbol{J_{RR}} = \begin{bmatrix} -a_1 \sin q_1 - a_2 \sin(q_1 + q_2) & -a_2 \sin(q_1 + q_2) \\ a_1 \cos q_1 + a_2 \cos(q_1 + q_2) & a_2 \cos(q_1 + q_2) \end{bmatrix} \tag{3.13}$$

The resolved rate algorithm is an elegant way to solve the inverse kinematics problem was proposed by Whitney, [45]. The main idea is to avoid solving for the inverse kinematics directly which is described in Eq. (3.7). Alternatively, the rates of the system are considered. The system is integrated one time step and the Jacobian is updated accordingly as described in Eq. (3.14) and (3.15).

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(q)\dot{\boldsymbol{x}} \tag{3.14}$$

$$q_{t+1} = q_0 + \dot{q}\Delta t \tag{3.15}$$

There are different methods to solve the Jacobian inverse. For example, the Damped Least Squares (DLS), Damped Least Squares with Singular Value Decomposition (DVS-DLS), Selectively Damped Least Squares (SDLS), Jacobian Transpose, and other extended versions of these methods, [46, 47, 48, 49, 50, 51]. Generally, Jacobian inverse methods suffer from singularities, involved matrix calculations, and high computational cost. In order to avoid singularities and increase the efficiency of the solutions, Pechev solved the inverse kinematics problem from a controls point of view, [52]. He constructed the transformation from Cartesian space to joint space in a feedback loop. Hence, the feedback inverse kinematics worked as a filter without relying on matrix inversion. Another set of inverse kinematics solutions are Newton's methods. Formulating the problem as minimization problem and seeking target configurations to return continuous smooth motion. Methods like Powell's method, Broyden's method, Fletcher, Goldfarb and Shanno (BFGS) method, [53]. Newton's methods are not easy to implement and are computationally costly, [54]. The method used in this work is the pseudo-inverse of the Jacobian for its simplicity. A feedback term was added in order to minimize the error.

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}(q)(\dot{\boldsymbol{x}}_r + \boldsymbol{K}\boldsymbol{e}) \tag{3.16}$$

where $\boldsymbol{e} = \boldsymbol{x_r} - \boldsymbol{x}$ is the error between the reference trajectory $\boldsymbol{x}_r$ and actual values of the task variables $\boldsymbol{x}$. The error is attenuated by a feedback proportional controller $\boldsymbol{K}$.

Simulation Results

In order to solve the problem of kinematic control, Eq. (3.16) is utilized. The reference velocity and path fed in to the pseudo-inverse kinematics of the system. The reference position is corrected by feeding back the position signal. Consequently, the resulting joint speeds are integrated to get the joint angles that will drive the forward kinematics to calculate the position of the end effector. The process is illustrated in Fig. 3.4.

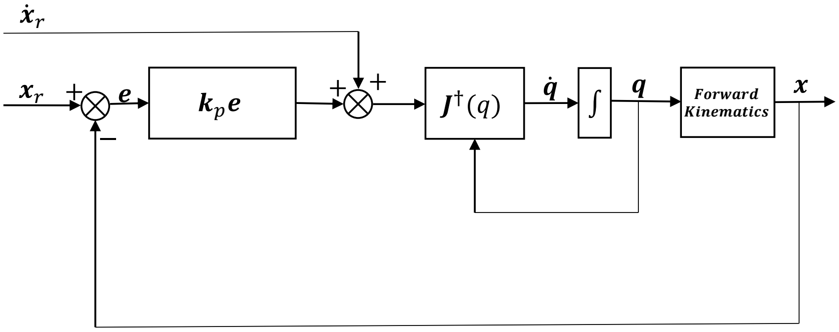
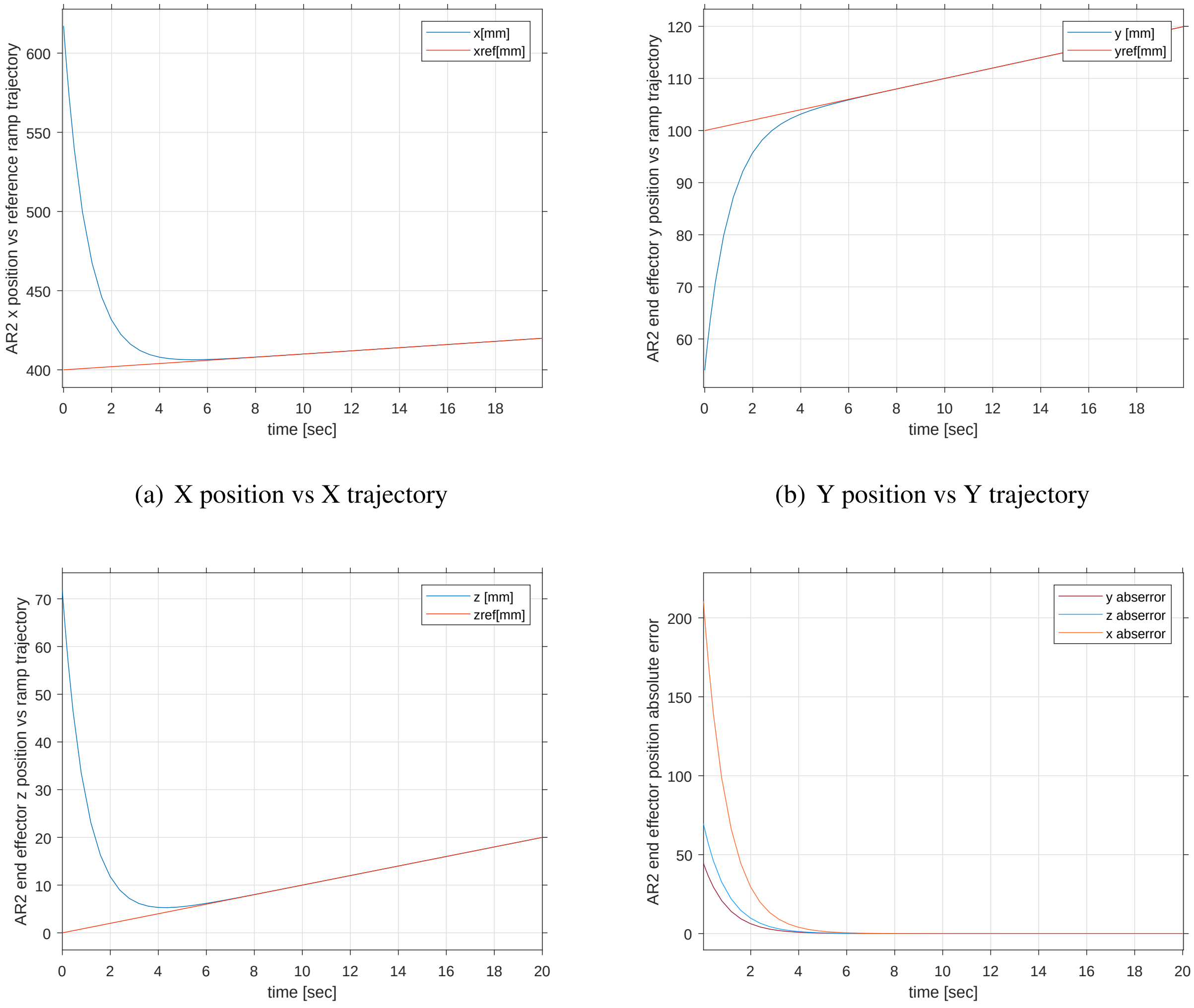


Figure 3.4: Resolved rate kinematic control

Two different inputs were used for simulation purposes. The first is a ramp velocity input and the other is a sine wave velocity reference. The system showed convergent behavior tracking these signals. The position of the end effector is plotted in Fig. 3.5 for the ramp input case. The error shows an exponential decrease. For the sine wave tracking, the system showed good performance even with a higher initial condition value as shown in Fig. 3.6(a).



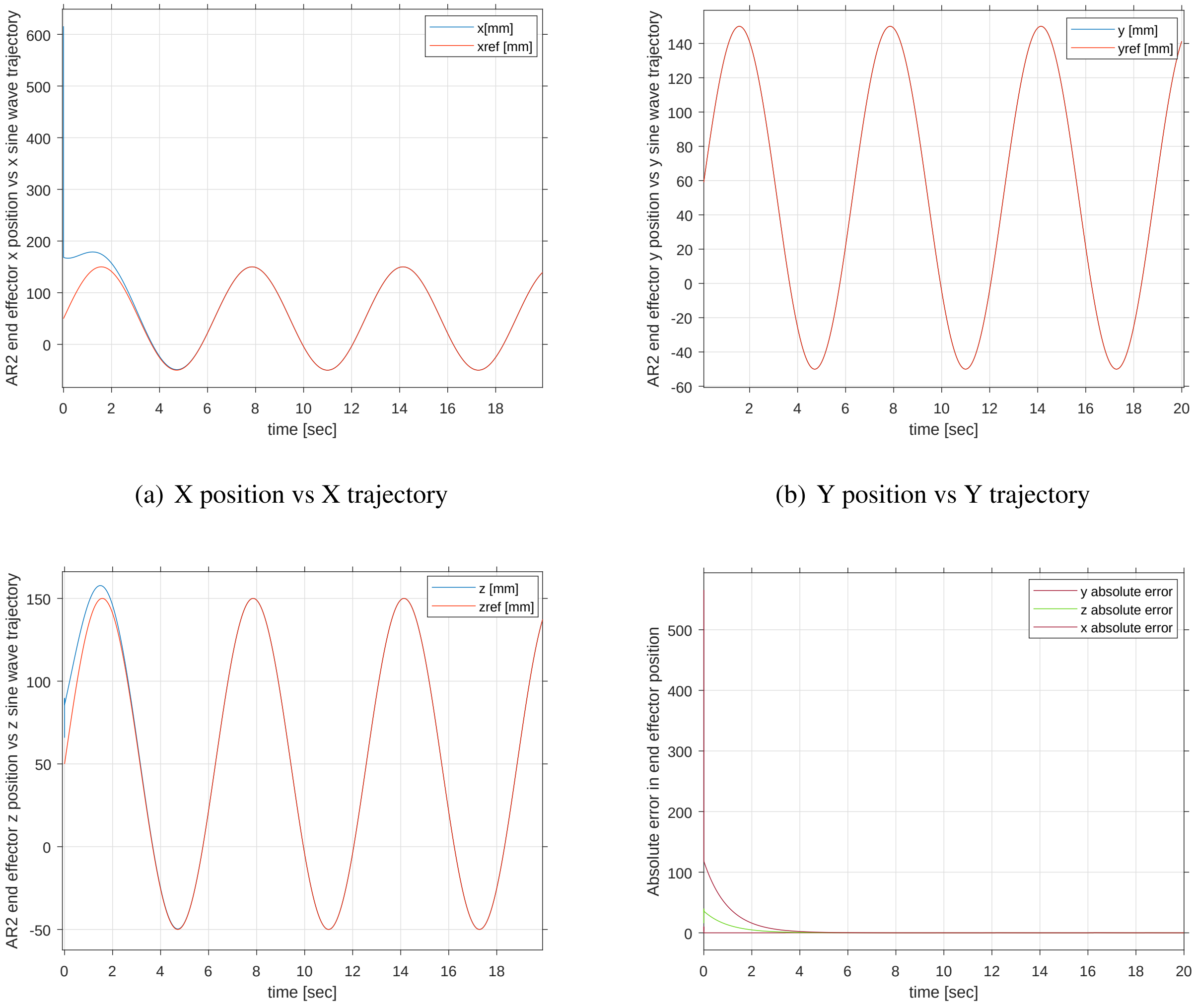
(a) X position vs X trajectory

(b) Y position vs Y trajectory

(c) Z posotion vs Z trajectory

(d) Z posotion vs Z trajectory

Figure 3.5: End effector position vs ramp reference using a proportional controller



(a) X position vs X trajectory

(b) Y position vs Y trajectory

(c) Z posotion vs Z trajectory

(d) Position absolute error

Figure 3.6: End effector position vs sine wave reference using a proportional controller

Solving the pose tracking problem is not as straight forward as path following. The rotational kinematic relations are of a nonlinear nature. Thus, a different pose following algorithm that propagates the attitude error kinematics is needed. The derivation is included for completeness as described by Siciliano, [55].

The desired and current rotation matrices of the end effector can be described as

$$\begin{aligned} \boldsymbol{R}_d &= [\boldsymbol{n}_d, \boldsymbol{s}_d, \boldsymbol{a}_d] \\ \boldsymbol{R}_e &= [\boldsymbol{n}_e, \boldsymbol{s}_e, \boldsymbol{a}_e] \end{aligned} \tag{3.17}$$

where $\boldsymbol{n}_d$, $\boldsymbol{s}_d$ and $\boldsymbol{a}_d$ are the column vectors of the rotation matrices. The orientation error can be written as

$$\delta\boldsymbol{R} = \boldsymbol{R}_d\boldsymbol{R}_e^T(q) \tag{3.18}$$

The error formulation described in Eq. (3.18) simplifies the pose tracking problem from 9 to only 3 elements. The latter definition can be compared with the rotation matrix

$$\boldsymbol{R}(\gamma, \boldsymbol{r}) = \begin{bmatrix} r_x^2(1-c_\gamma)+c_\gamma & r_xr_y(1-c_\gamma)-r_zs_\gamma & r_xr_z(1-c_\gamma)+r_ys_\gamma \\ r_xr_y(1-c_\gamma)+r_zs_\gamma & r_y^2(1-c_\gamma)+c_\gamma & r_yr_z(1-c\gamma)-r_xs_\gamma \\ r_xr_z(1-c_\gamma)-r_ys_\gamma & r_yr_z(1-c_\gamma)+r_xs_\gamma & r_z^2(1-c_\gamma)+c_\gamma \end{bmatrix} \tag{3.19}$$

where $\boldsymbol{r}$ is the principle vector of rotation and γ is the principle angle of rotation. The error can be written as

$$e_o = \boldsymbol{r}sin(\gamma) \tag{3.20}$$

By comparing the error matrix in Eq. (3.18) and the matrix in Eq. (3.19), the following definition of the orientation error and its derivative can be written

$$e_o = \frac{1}{2}(\boldsymbol{n}_e(q)\times\boldsymbol{n}_d + \boldsymbol{s}_e(q)\times\boldsymbol{s}_d + \boldsymbol{a}_e(q)\times\boldsymbol{a}_d)\dot{e}_o = \boldsymbol{L}^T\omega_d - \boldsymbol{L}\omega_e \tag{3.21}$$

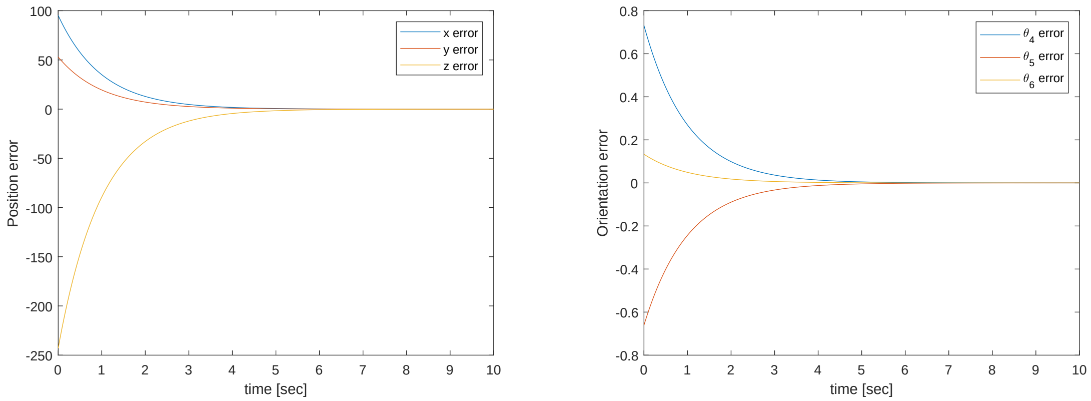
where

$$\boldsymbol{L} = -\frac{1}{2}(\boldsymbol{S}(n_d)\boldsymbol{S}(n_e) + \boldsymbol{S}(s_d)\boldsymbol{S}(s_e) + \boldsymbol{S}(a_d)\boldsymbol{S}(a_e)) \tag{3.22}$$

The system of equations to be solved for pose tracking can be written as

$$\dot{\boldsymbol{e}} = \begin{bmatrix} \dot{\boldsymbol{e}}_p \\ \dot{\boldsymbol{e}}_o \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_d - \boldsymbol{J}_p(q)\dot{\boldsymbol{q}} \\ \boldsymbol{L}^T\boldsymbol{\omega}_d - \boldsymbol{L}\boldsymbol{J}_o(q)\dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_d \\ \boldsymbol{L}^T\boldsymbol{\omega}_d \end{bmatrix} - \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L} \end{bmatrix} \boldsymbol{J}\dot{\boldsymbol{q}} \tag{3.23}$$

The system of Eq. (3.23) is integrated using MATLAB ode45 over a time span of 10 seconds. The pose error showed a converging behavior as shown in Fig. 3.7



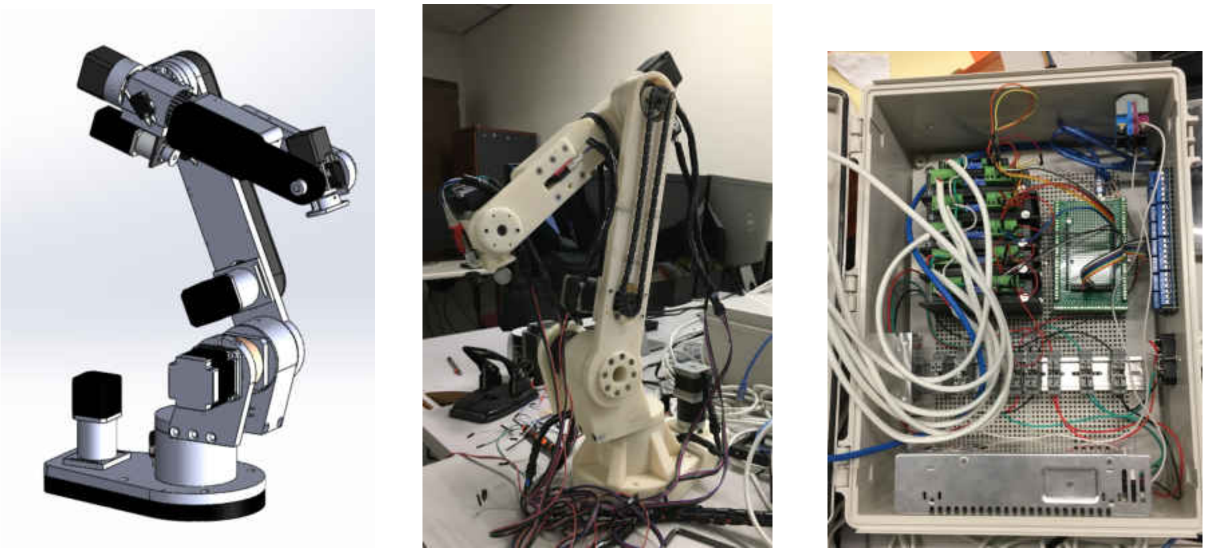
(a) AR2 position tracking absolute error

(b) AR2 orientation tracking absolute error

Figure 3.7: AR2 pose tracking absolute error

Experimental Results

The selection criteria of the robotic manipulator for ROME are precision and cost efficiency driven. Relaxing constraints like grip force and payload weight, the selection process was mainly thrusted by the manipulator mass, repeatablility and reachability. Through a market analysis, it was found that industrial manipulators were too expensive and hobbyist robotic manipulator kits must be considered. In addition, manipulators that utilize stepper motors in favor of servo motors were preferred because of its higher precision. Following a budget plan, the AR2 robotic manipulator designed by Chris Annin was utilized to generate the preliminary results and proof of concept of this project. AR2, see Fig. 3.8 , has 6-DOF, easy to use and program and costs less than $\$1000$. The price range is very competitive compared to its available counterparts in the market. The manipulator can be manufactured by 3D printing or by machining aluminum. The main hardware board used to control the manipulator is an Arduino Mega 2650. The complete assembly of the robotic arm can be seen in Fig. 3.8(b). The electrical enclosure setup for the robotic manipulator has to be attached to it and can be seen in Fig. 3.8(c).



(a) AR2 CAD assembly

(b) Completed AR2 assembly

(c) Electrical enclosure for AR2

Figure 3.8: AR2 manipulator CAD, 3D printed models and electrical enclosure

The experiments performed using AR2 robotic manipulator took place only in the vertical z direction of the end effector. In order to achieve accurate results, three of the six available joints are utilized and the rest are locked. The time span of all the experiments is 60 seconds.

The first experiment performed with a ramp reference position and a constant velocity described as follows

$$z(t) = \frac{80}{60}t \qquad \dot{z}(t) = \frac{80}{60} \tag{3.24}$$

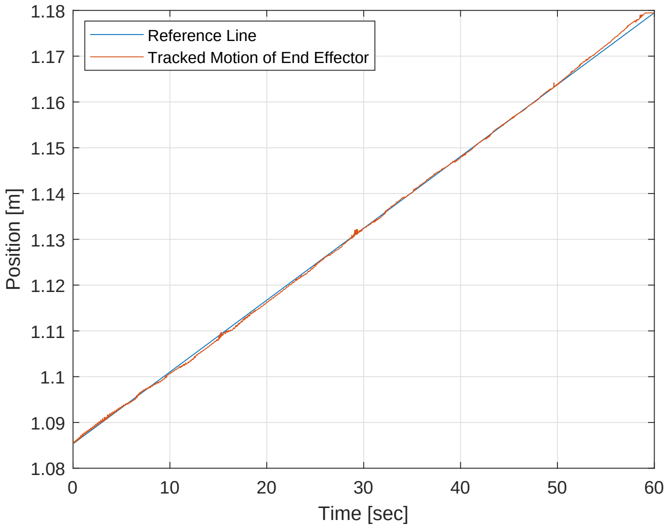
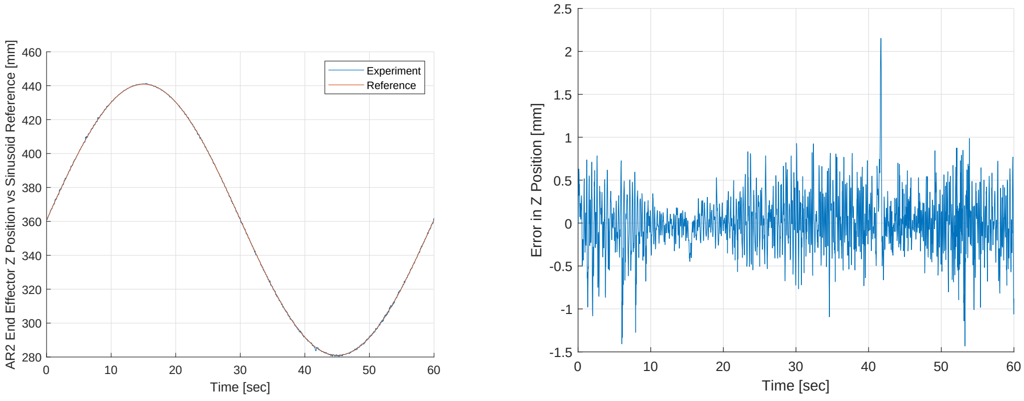


Figure 3.9: AR2 constrained motion in Z direction: ramp trajectory with open loop control

Running a simulation of the AR2 robot with Eq. (3.24) as a reference input, the results have minimal error as shown in Fig. 3.9. For this time span, feedback control is not necessary to achieve satisfactory results. For a nonlinear reference path, the simple closed loop resolved rate algorithm described in Fig. 3.4 is used to reduce numerical errors. The sinusoidal reference input in Eq. (3.25) and an orbit equation is used for simulation. The results are shown in Fig. 3.10 and 3.11.

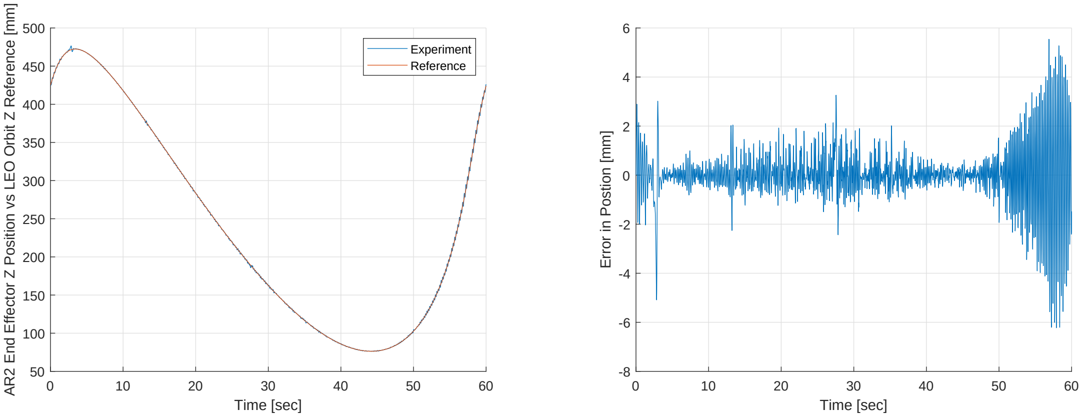
$$z(t) = 80\sin\left(\frac{2\pi}{60}t\right) \qquad \dot{z}(t) = \frac{160\pi}{60}\cos\left(\frac{2\pi}{60}t\right) \tag{3.25}$$



(a) AR2 experiment following as sinusoidal Wave: Z position

(b) Absolute error in position vs time

Figure 3.10: AR2 constrained motion in Z direction: Sinusoid experiment using PI controller



(a) AR2 experiment following LEO orbit: Z Position

(b) Absolute error in position vs time

Figure 3.11: AR2 constrained motion in Z direction: LEO experiment using PID controller

Using a Proportional Integral (PI) controller improved the performance and bounded the error to an average of 1mm as shown in Fig. 3.10(b). In addition, the error reaches its peak around the inflection points of the sinusoidal wave, partly due to the pseudo-inverse solution of the Jacobian.

CHAPTER 4: ROME MOBILE MANIPULATOR

ROME is a mobile manipulator composed of a three wheeled omnidirectional ground vehicle with a 6-DOF robotic manipulator fixed on top of as shown in Fig. 4.1. The system has 9-DOF in total. The new challenge working with such a system is the redundancy problem which occurs when the number of actuators exceed the number of physical degrees of freedom of the system. A simple way to deal with redundancy is to use the kinematic models for the ground vehicle and the robotic manipulator developed in the previous chapters. The main strategy is to make the ground vehicle follow a reference trajectory and prompt the ground vehicle and the robotic manipulator to correct for the errors. The velocity and position in the (x_r, y_r) plane along with the yaw rate and yaw angle ψ_r is provided as a reference input for the ground vehicle. The vertical velocity and the vertical motion z_r is handled solely by the robotic manipulator.

The process is shown in Fig. 4.2. The symbols ”GV FK” and ”AR2 FK” stand for the ground vehicle forward kinematics and AR2 robotic manipulator forward kinematics simultaneously. Path following problem is solved following the previously mentioned strategy. The simulation results for path following are shown in Fig. 4.3. Moreover, the pose tracking simulation results is presented in Fig. 4.4. The system converges to the desired path in less than one second and to the desired orientation in less than 5 seconds. The control law used for the mobile manipulator is that of the ground vehicle in Eq. (2.17) and the robotic manipulator in Eq. (3.23).

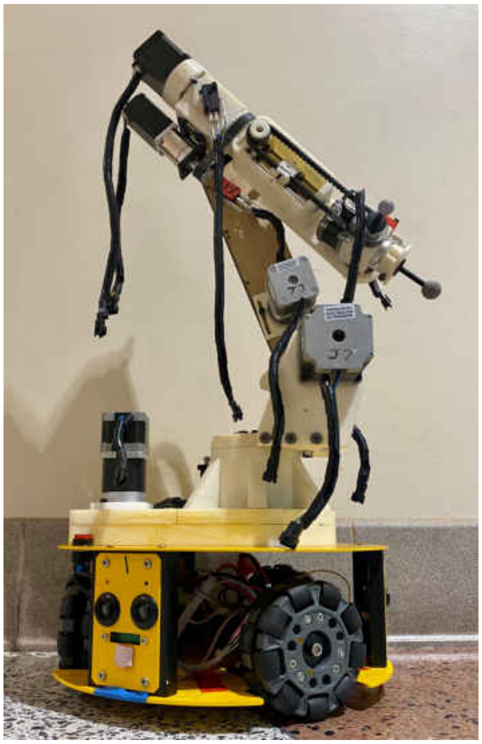


Figure 4.1: ROME hardware assembly

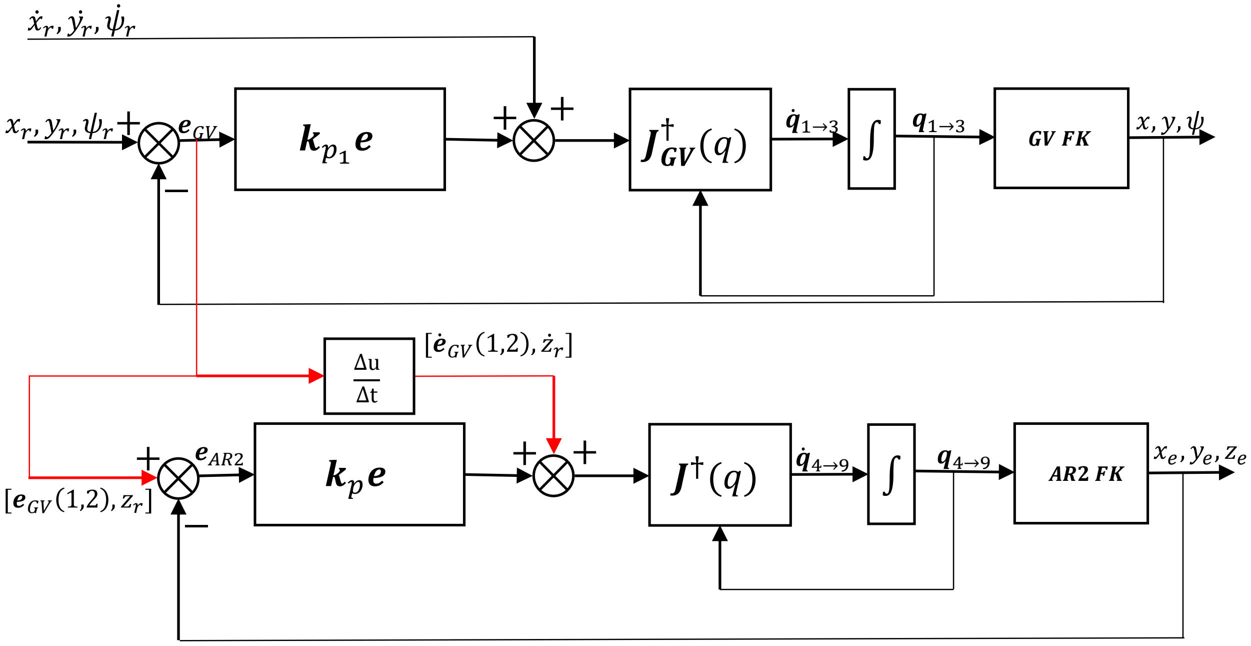
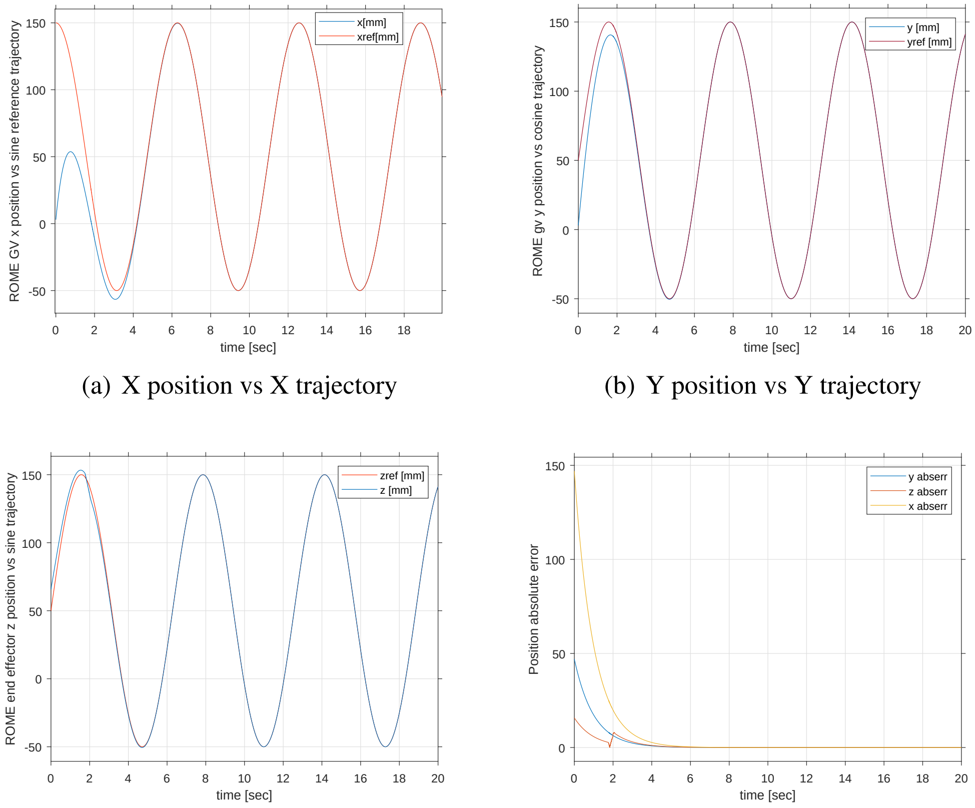


Figure 4.2: ROME closed loop resolved rate block diagram



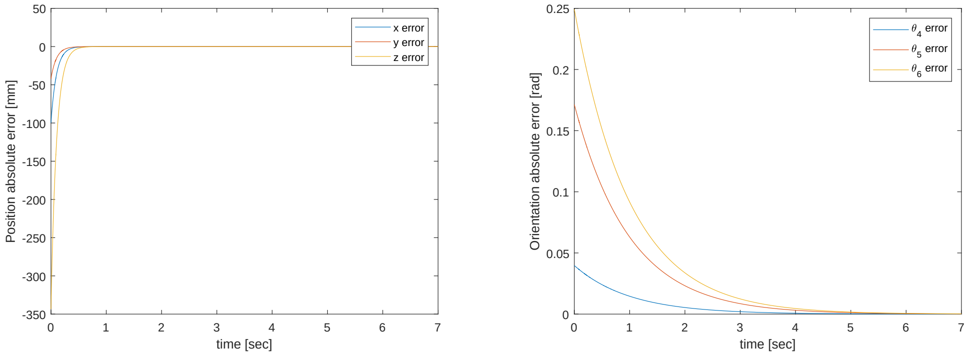
(a) X position vs X trajectory

(b) Y position vs Y trajectory

(c) Z posotion vs Z trajectory

(d) Position absolute error

Figure 4.3: End effector position vs reference position using a Proportional controller



(a) ROME position tracking absolute error

(b) ROME orientation tracking absolute error

Figure 4.4: ROME pose tracking absolute error

Application to Astrodynamics

The two-body problem can be directly applied to satellites motion, or the International Space Station orbiting earth, [56]. ROME is used to emulate the orbital motion in a laboratory environment. The solution of the two-body problem is the Cartesian trajectory to be followed by the end effector. Consider the classical two body motion differential equation

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} \tag{4.1}$$

where r is the position vector from one body to the other and μ is the gravitational constant. Lagrange Gibbs (F & G) method offers a closed form solution for the two body problem. At a given instant, if the position and velocity of an orbiting body are known, then the evolution of the position and velocity with time can be found in terms of the initial conditions. Consider the position and velocity vectors

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{x}\hat{\boldsymbol{p}} + \boldsymbol{y}\hat{\boldsymbol{q}} \\ \boldsymbol{v} &= \dot{\boldsymbol{r}} = \dot{\boldsymbol{x}}\hat{\boldsymbol{p}} + \dot{\boldsymbol{y}}\hat{\boldsymbol{q}} \end{aligned} \tag{4.2}$$

for the initial conditions at $t = t_0$

$$\begin{aligned} \boldsymbol{r_0} &= \boldsymbol{x}_0\hat{\boldsymbol{p}} + \boldsymbol{y}_0\hat{\boldsymbol{q}} \\ \boldsymbol{v_0} &= \dot{\boldsymbol{x}}_0\hat{\boldsymbol{p}} + \dot{\boldsymbol{y}}_0\hat{\boldsymbol{q}} \end{aligned} \tag{4.3}$$

From the definition of angular momentum

$$\boldsymbol{h} = \boldsymbol{r}_0 \times \boldsymbol{v}_0 = \begin{bmatrix} \hat{\boldsymbol{p}} & \hat{\boldsymbol{q}} & \hat{\boldsymbol{w}} \\ \boldsymbol{x}_0 & \boldsymbol{y}_0 & 0 \\ \dot{\boldsymbol{x}}_0 & \dot{\boldsymbol{y}}_0 & 0 \end{bmatrix} = \hat{\boldsymbol{w}}(\boldsymbol{x}_0\dot{\boldsymbol{y}}_0 - \boldsymbol{y}_0\dot{\boldsymbol{x}}_0) \tag{4.4}$$

A description of the $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ vectors in terms of the initial conditions can be calculated and substituted into Eq. 4.2 to get

$$\begin{aligned} \boldsymbol{r} &= \frac{\boldsymbol{x}\dot{\boldsymbol{y}}_0 - \boldsymbol{y}\dot{\boldsymbol{x}}_0}{h}\boldsymbol{r}_0 + \frac{-\boldsymbol{x}\boldsymbol{y}_0 + \boldsymbol{y}\boldsymbol{x}_0}{h}\boldsymbol{v}_0 \\ \boldsymbol{v} &= \frac{\dot{\boldsymbol{x}}\dot{\boldsymbol{y}}_0 - \boldsymbol{y}\dot{\boldsymbol{x}}_0}{h}\boldsymbol{r}_0 + \frac{-\dot{\boldsymbol{x}}\boldsymbol{y}_0 + \dot{\boldsymbol{y}}\boldsymbol{x}_0}{h}\boldsymbol{v}_0 \end{aligned} \tag{4.5}$$

which can be rewritten in the form

$$\begin{aligned} \boldsymbol{r} &= f\boldsymbol{r}_0 + g\boldsymbol{v}_0 \\ \boldsymbol{v} &= \dot{f}\boldsymbol{r}_0 + \dot{g}\boldsymbol{v}_0 \end{aligned} \tag{4.6}$$

where f and g are given by

$$\begin{aligned} f &= \frac{\boldsymbol{x}\dot{\boldsymbol{y}}_0 - \boldsymbol{y}\dot{\boldsymbol{x}}_0}{h}\boldsymbol{r}_0 \\ g &= \frac{-\boldsymbol{x}\boldsymbol{y}_0 + \boldsymbol{y}\boldsymbol{x}_0}{h}\boldsymbol{v}_0 \end{aligned} \tag{4.7}$$

and their time derivatives are

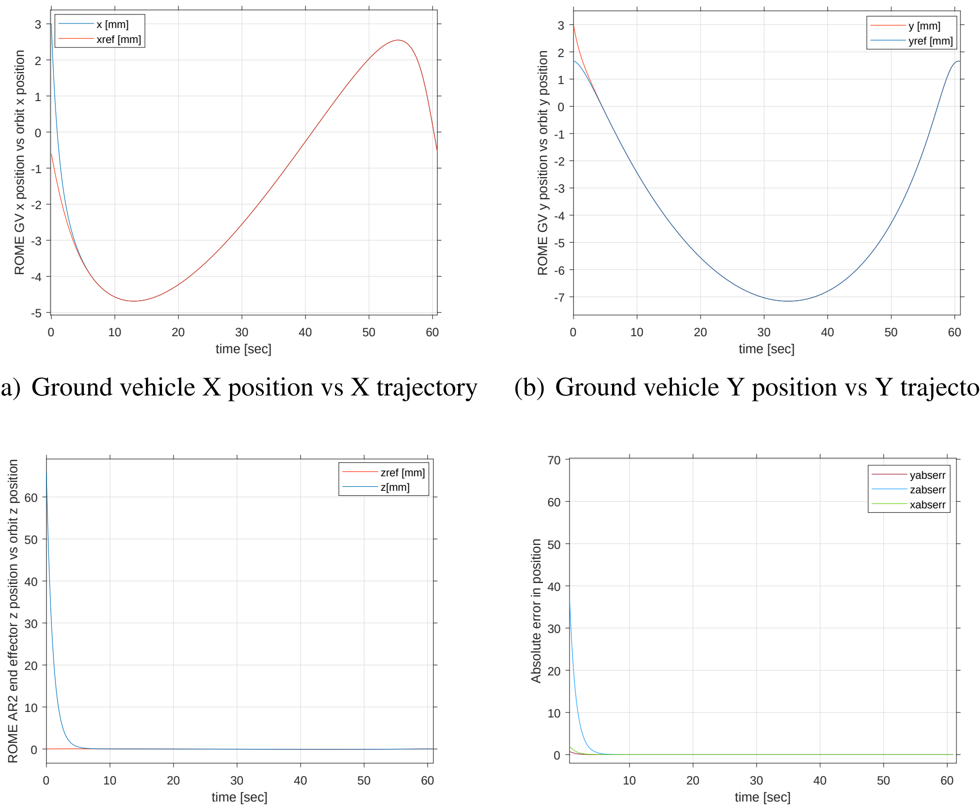
$$\begin{aligned} \dot{f} &= \frac{\dot{\boldsymbol{x}}\dot{\boldsymbol{y}}_0 - \boldsymbol{y}\dot{\boldsymbol{x}}_0}{h} \\ \dot{g} &= \frac{-\dot{\boldsymbol{x}}\boldsymbol{y}_0 + \dot{\boldsymbol{y}}\boldsymbol{x}_0}{h} \end{aligned} \tag{4.8}$$

To have a proper scale to execute experiments in a confined laboratory environment, the orbit equation and the initial conditions are normalized. The earth equatorial radius DU is used to normalize the distance and the solar second TU is used to normalize the time. They are defined as follows

$$DU = 6378.137 \times 10^3, \quad TU = \sqrt{\frac{DU^3}{\mu}}, \quad VU = \frac{DU}{TU} \tag{4.9}$$

where, $M$is the mass of the earth, the gravitational constant $\mu = 1$ for the canonical units and VU is the normalized velocity. The equation was solved using the Lagrange Gibbs (F&G) solution for a Low Earth Orbit (LEO) with the following initial position after normalization. The solution of the orbit is shown in Fig. 4.6(a)

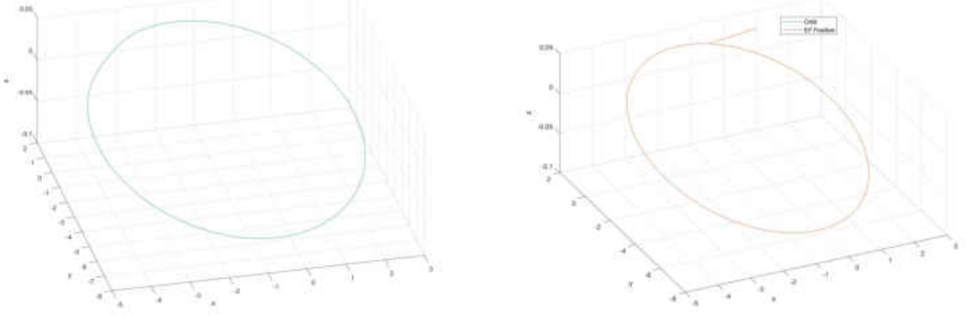
$$\begin{aligned} Position &= [-0.5985 \quad 1.6634 \quad 0.0279] \\ Velocity &= [-0.9523 \quad -0.0639 \quad 0.0110] \end{aligned} \tag{4.10}$$



(a) Ground vehicle X position vs X trajectory (b) Ground vehicle Y position vs Y trajectory

(c) AR2 end effector Z position vs Z trajectory (d) ROME absolute position error

Figure 4.5: ROME following an elliptic orbit in a decoupled fashion



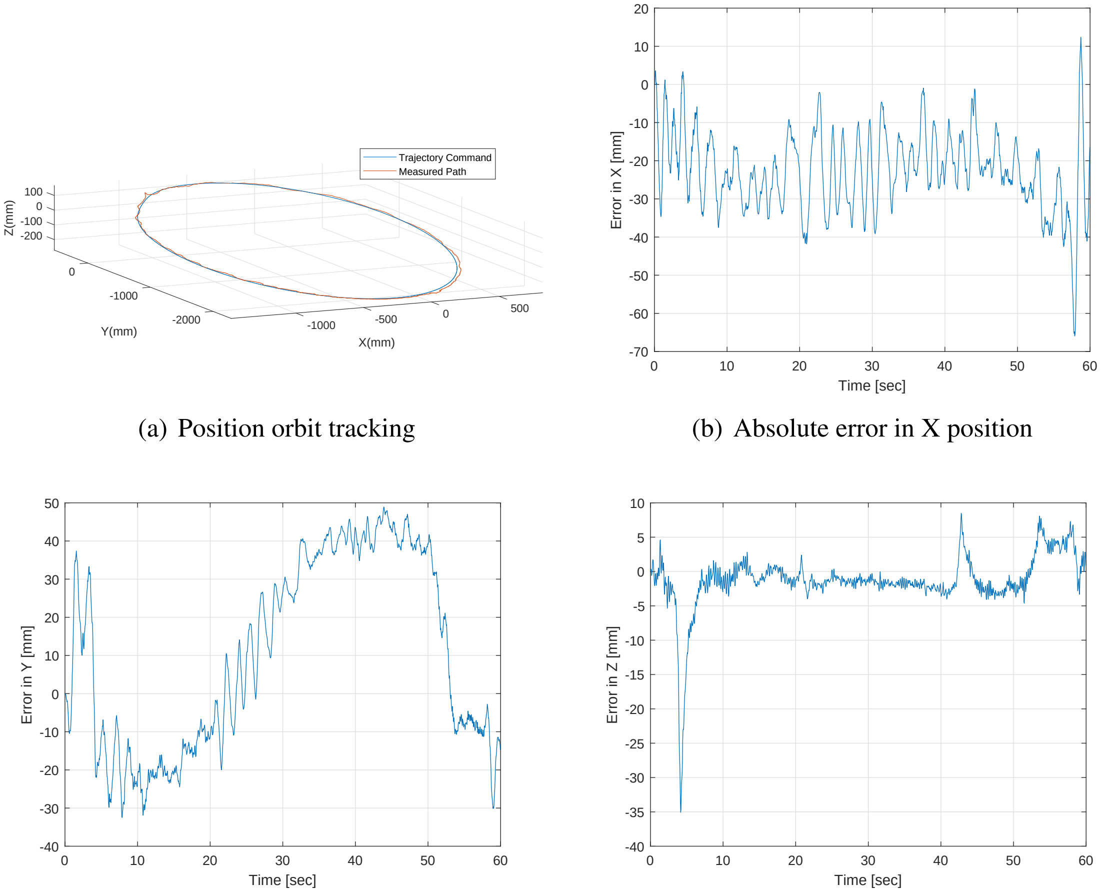
(a) LEO orbit solution using Lagrange Gibbs method

(b) ROME decoupled simulation

Figure 4.6: 3D plots for ROME following an elliptic orbit

The orbit equation is solved in the Simulink environment and used as a trajectory for simulation. The orbit tracking simulation is performed in a similar manner as the sinusoidal trajectory tracking. The velocities $(\dot{x}_r, \dot{y}_r)$ and the position (x_r, y_r) are used as an input to the ground vehicle. Further, the velocity and position $(z_r, \dot{z}_r)$ are used to drive the robotic manipulator simultaneously. The simulation time is calculated based on the calculated orbit period $t_f = 60.8844$ seconds. The simulation showed that the system is able to track an elliptic orbit as presented in Fig. 4.6(b). The error converges to zero as shown in Fig. 4.5.

The same orbit is used for ROME orbit following experiment. Following the same procedure, the vehicle follows the planar position input and the end effector follows the velocity and position reference in the z_r direction. The robotic system successfully followed the reference orbit with some oscillations as shown in Fig. 4.7.



(a) Position orbit tracking

(b) Absolute error in X position

(c) Absolute error in Y position

(d) Absolute error in Z position

Figure 4.7: Error in ground plane motion produced by ground vehicle

CHAPTER 5: CONCLUSION AND FUTURE WORK

A mobile manipulator was developed to emulate orbital motion in a lab environment. The robot cost did not exceed $\$1500$. The forward kinematics of the ground vehicle, robotic manipulator and ROME were derived. The resolved rate of motion for the robotic manipulator and the mobile manipulator were carried out using Simulink. The simulation results showed convergent behavior. The experimental results of the closed loop kinematics of the ground vehicle show good accuracy compared to the model. The solution of the two-body problem is solved to get an orbit trajectory. The robot followed the elliptical trajectory showing good accuracy with some vibrations. The vibrations are due to imperfections in the floor and the light weight of the ground vehicle. Moreover, the vibrations in the ground vehicle need to be reduced by using higher quality motors. The system can be extended to simulate and test various space missions and control algorithms like docking maneuvers and servicing missions. The current system is a good preliminary proof of concept for the idea of emulating orbital motion using mobile manipulators.

For future work, the ground vehicle Chassis will be designed to have less vibrations to include a compartment design for the electronics and the batteries. The robotic manipulator power supply will be redesigned to be portable and lightweight. The dynamics with closed loop feedback control will be used to eliminate the vibrations from the system. Model Reference Adaptive Control (MRAC) will be used to follow and imitate the dynamics of orbital motion and other dynamical systems. The hardware architecture will be investigated for more stability and robustness. Furthermore, the gravity compensation problem will be addressed. In order to maintain equilibrium while carrying out a task, the gravity vector has to be compensated using mechanisms or feedback closed loop control. Moreover, the gravity manipulation plays a major role in forcing the robotic manipulator to simulate space motion. The gravity effects of a robotic manipulator can be found by formulating the system dynamics.

LIST OF REFERENCES

[1] Corke, P., *Robotics, vision and control: fundamental algorithms In MATLAB® second, completely revised*, Vol. 118, Springer, 2017.

[2] Annin, C., “https://www.anninrobotics.com,” .

[3] Kwok-Choon, S., Buchala, K., Blackwell, B., Lopresti, S., Wilde, M., and Go, T., “Design, Fabrication, and Preliminary Testing of Air-Bearing Test Vehicles for the Study of Autonomous Satellite Maneuvers,” 2018.

[4] Mietner, C., “European Proximity Operations Simulator 2.0 (EPOS) - A Robotic-Based Rendezvous and Docking Simulator,” *Journal of Large-Scale Research Facilities JLSRF*, Vol. 3, 04 2017.

[5] Ananthakrishnan, S., Teders, R., and Alder, K., “Role of estimation in real-time contact dynamics enhancement of space station engineering facility,” *IEEE Robotics & Automation Magazine*, Vol. 3, No. 3, 1996, pp. 20–28.

[6] Piedboeuf, J.-C., De Carufel, J., Aghili, F., and Dupuis, E., “Task verification facility for the Canadian special purpose dextrous manipulator,” *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, Vol. 2, IEEE, 1999, pp. 1077–1083.

[7] Bell, R., Collins, J., Wertz, J., and Hansen, L. J., “Hardware-in-the Loop Tests of an Autonomous GN&C System for On-orbit Servicing,” *AIAA Space 2003 Conference & Exposition*, 2003, p. 6372.

[8] Bai, X., Davis, J., Doebbler, J., Turner, J., and Junkins, J. L., "Dynamics, control and simulation of a mobile robotics system for 6-dof motion emulation," *World Congress on Engineering and Computer Science, San Francisco, CA*, 2007.

[9] Foust, R. C., Lupu, E. S., Nakka, Y. K., Chung, S.-J., and Hadaegh, F. Y., "Ultra-Soft Electromagnetic Docking with Applications to In-Orbit Assembly," 2018.

[10] Papadopoulos, E., Paraskevas, I., Flessa, T., Nanos, K., Rekleitis, Y., and Kontolatis, I., "The NTUA Space Robot Simulator: Design & Results," 11 2008.

[11] Cavalieri, K. A., Macomber, B., Moody, C., Probe, A., and Junkins, J. L., "Laboratory Experiments Supporting Autonomous Space Debris Mitigation," *Proc. of the 36th annual AAS rocky mountain section guidance and control conference*, Vol. 149, 2013, pp. 3–16.

[12] Mao, Q. and Wang, S., "Reachable Relative Motion Design of Space Manipulator Actuated Microgravity Platform," *Journal of Spacecraft and Rockets*, 2018, pp. 1–13.

[13] Arakelian, V., "Gravity compensation in robotics," *Advanced Robotics*, Vol. 30, No. 2, 2016, pp. 79–96.

[14] Arakelian, V., "The history of the creation and development of hand-operated balanced manipulators (HOBM)," *International Symposium on History of Machines and Mechanisms*, Springer, 2004, pp. 347–356.

[15] Fukushima, E. F., Debenest, P., Tojo, Y., Takita, K., Freese, M., RADRICH, H., and HIROSE, S., "Teleoperated buggy vehicle and weight balanced arm for mechanization of mine detection and clearance tasks," *Proc. of the HUDEM*, 2005, pp. 58–63.

[16] Tzafestas, S. G., *Introduction to mobile robot control*, Elsevier, 2013.

[17] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D., *Introduction to autonomous mobile robots*, MIT press, 2011.

[18] Holmberg, R. and Khatib, O., "Development and control of a holonomic mobile robot for mobile manipulation tasks," *The International Journal of Robotics Research*, Vol. 19, No. 11, 2000, pp. 1066–1074.

[19] Batavia, P. H. and Nourbakhsh, I., "Path planning for the Cye personal robot," *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, Vol. 1, IEEE, 2000, pp. 15–20.

[20] Blackwell, M., "The uranus mobile robot," Tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1990.

[21] Williams, R. L., Carter, B. E., Gallina, P., and Rosati, G., "Dynamic model with slip for wheeled omnidirectional robots," *IEEE transactions on Robotics and Automation*, Vol. 18, No. 3, 2002, pp. 285–293.

[22] Watanabe, K., Shiraishi, Y., Tang, J., Fukuda, T., and Tzafestas, S. G., "Autonomous Control for an Omnidirectional Mobile Robot with Feedback Control System," *Advances in Intelligent Autonomous Systems*, Springer, 1999, pp. 289–308.

[23] Choi, B. and Sreenivasan, S., "Gross motion characteristics of articulated mobile robots with pure rolling capability on smooth uneven surfaces," *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 2, 1999, pp. 340–343.

[24] Rajagopalan, R., "A generic kinematic formulation for wheeled mobile robots," *Journal of Robotic Systems*, Vol. 14, No. 2, 1997, pp. 77–91.

[25] Shekhar, S., "Wheel rolling constraints and slip in mobile robots," *Proceedings of International Conference on Robotics and Automation*, Vol. 3, IEEE, 1997, pp. 2601–2607.

[26] Balakrishna, R. and Ghosal, A., "Modeling of slip for wheeled mobile robots," *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 1, 1995, pp. 126–132.

[27] Scheding, S., Dissanayake, G., Nebot, E. M., and Durrant-Whyte, H., "An experiment in autonomous navigation of an underground mining vehicle," *IEEE Transactions on robotics and Automation*, Vol. 15, No. 1, 1999, pp. 85–95.

[28] Tzafestas, S. G., "Mobile robot control and navigation: A global overview," *Journal of Intelligent & Robotic Systems*, Vol. 91, No. 1, 2018, pp. 35–58.

[29] Ruehl, S. W., Konle, A., Roennau, A., and Dillmann, R., "SIMULATION OF ON-ORIBIT-SERVICING TASKS USING A MOBILE MANIPULATOR AND A PHYSICS SIMULATION," .

[30] Peterson, L., Austin, D., and Kragic, D., "High-level control of a mobile manipulator for door opening," *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, Vol. 3, IEEE, 2000, pp. 2333–2338.

[31] Hamner, B., Koterba, S., Shi, J., Simmons, R., and Singh, S., "An autonomous mobile manipulator for assembly tasks," *Autonomous Robots*, Vol. 28, No. 1, 2010, pp. 131.

[32] Nagatani, K. and Yuta, S., "Designing strategy and implementation of mobile manipulator control system for opening door," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 3, IEEE, 1996, pp. 2828–2834.

[33] Chitta, S., Cohen, B., and Likhachev, M., "Planning for autonomous door opening with a mobile manipulator," *2010 IEEE International Conference on Robotics and Automation*, IEEE, 2010, pp. 1799–1806.

[34] Pin, F. G. and Culioli, J.-C., "Optimal positioning of combined mobile platform-manipulator systems for material handling tasks," *Journal of intelligent and Robotic Systems*, Vol. 6, No. 2-3, 1992, pp. 165–182.

[35] Jain, A. and Kemp, C. C., “EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces,” *Autonomous Robots*, Vol. 28, No. 1, 2010, pp. 45.

[36] Agah, A. and Tanie, K., “Human interaction with a service robot: Mobile-manipulator handing over an object to a human,” *Proceedings of International Conference on Robotics and Automation*, Vol. 1, IEEE, 1997, pp. 575–580.

[37] Tomizawa, T., Ohya, A., and Yuta, S., “Remote book browsing system using a mobile manipulator,” *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, Vol. 1, IEEE, 2003, pp. 256–261.

[38] Bort, L. and del Pobil, A. P., “Using speech to guide a mobile robot manipulator,” *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, Vol. 4, IEEE, 2000, pp. 2356–2361.

[39] Osumi, H., Terasawa, M., and Nojiri, H., “Cooperative control of multiple mobile manipulators on uneven ground,” *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, Vol. 4, IEEE, 1998, pp. 3198–3203.

[40] Stroupe, A., Huntsberger, T., Okon, A., Aghazarian, H., and Robinson, M., “Behavior-based multi-robot collaboration for autonomous construction tasks,” *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2005, pp. 1495–1500.

[41] Yamamoto, Y. and Yun, X., “Coordinating locomotion and manipulation of a mobile manipulator,” *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, IEEE, 1992, pp. 2643–2648.

[42] Chen, M. and Zalzala, A. M., “A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration,” *Journal of Robotic Systems*, Vol. 14, No. 7, 1997, pp. 529–544.

[43] Tsai, C.-C., Jiang, L.-B., Wang, T.-Y., and Wang, T.-S., “Kinematics control of an omnidirectional mobile robot,” *Proceedings of 2005 CACS Automatic Control Conference*, Citeseer, 2005, pp. 13–18.

[44] Jazar, R. N., *Theory of applied robotics: kinematics, dynamics, and control*, Springer Science & Business Media, 2010.

[45] Whitney, D. E., “Resolved motion rate control of manipulators and human prostheses,” *IEEE Transactions on man-machine systems*, Vol. 10, No. 2, 1969, pp. 47–53.

[46] Balestrino, A., De Maria, G., and Sciavicco, L., “Robust control of robotic manipulators,” *IFAC Proceedings Volumes*, Vol. 17, No. 2, 1984, pp. 2435–2440.

[47] Wolovich, W. A. and Elliott, H., “A computational technique for inverse kinematics,” *Decision and Control, 1984. The 23rd IEEE Conference on*, Vol. 23, IEEE, 1984, pp. 1359–1363.

[48] Baillieul, J., “Kinematic programming alternatives for redundant manipulators,” *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, Vol. 2, IEEE, 1985, pp. 722–728.

[49] Wampler, C. W., “Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods,” *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 16, No. 1, 1986, pp. 93–101.

[50] Nakamura, Y. and Hanafusa, H., “Inverse kinematic solutions with singularity robustness for robot manipulator control,” *Journal of dynamic systems, measurement, and control*, Vol. 108, No. 3, 1986, pp. 163–171.

[51] Buss, S. R. and Kim, J.-S., “Selectively damped least squares for inverse kinematics,” *Journal of Graphics tools*, Vol. 10, No. 3, 2005, pp. 37–49.

[52] Pechev, A. N., “Inverse kinematics without matrix inversion,” *Proceedings of IEEE International Conference on Robotics and Automation*, 2008, pp. 2005–2012.

[53] Fletcher, R., *Practical methods of optimization*, John Wiley & Sons, 2013.

[54] Aristidou, A. and Lasenby, J., “Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver,” 09 2009.

[55] Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G., *Robotics: modelling, planning and control*, Springer Science & Business Media, 2010.

[56] Curtis, H. D., *Orbital mechanics for engineering students*, Butterworth-Heinemann, 2013.