National University of Singapore
Facility of Engineering
Department of Mechanical Engineering

Modeling and Constrained Control of the Autonomous Robot

ME5402 CA 2

Student Name: Sun Zhanhong A0225282J
e0575916@u.nus.edu
Zhou Kunran A0225441N
e0576075@u.nus.edu
Cao Yuhong A0219126J
e0546139@u.nus.edu
Supervisor: Prof. GE, Shuzhi Sam

ME5402/EE5106 ADVANCED ROBOTICS

Group Projects - Honor Pledge

Year	Semester	Project Choice
2020-2021	2	Modeling and Constrained Control of the Autonomous Robot

We, the following students of NUS, upon our honor, hereby confirm that we have neither received nor given any unauthorized help on the ME5402/EE5106 group project carried out by us.

The project reports reflect truly our own efforts. In all cases where material from other sources such as books, articles, notes and websites have been used, we have taken care to provide clear and unambiguous references to the same.

We confirm that we have not provided unauthorized help to other groups doing the same project. Furthermore, we also confirm that we will not pass on our research materials, report and presentation materials to other students who may take this module in later semesters.

In addition, this project report has been prepared and submitted by us only as a part of an academic exercise. Its contents are not meant for publication in any manner.

Sl.No.	Matric No.	Name	Question	Signature	Date
1.	A0225282J	Sun Zhanhong	1-2,1-4	Sun Zhanhong	2021.4.16
2.	A0225441N	Zhou Kunran	1-1	Zhou Kunran	2021.4.16
3.	A0219126J	Cao Yuhong	1-3	Cao Yuhong	2021.4.16

Contents

1. Literature Review **1**
1.1. Introduction . . . 1
1.2. Autonomous Robot Design . . . 1
1.3. Virtual Robot Simulation . . . 2
1.4. Constrained Control . . . 3

2. Lagrange-Euler Equations Derivation **5**
2.1. Robot Arm . . . 5
2.2. Autonomous Robot . . . 8

3. CAD Model and Model Import **12**
3.1. CAD model of the autonomous robot . . . 12
3.2. Import Model to Matlab . . . 12
3.2.1. XML File . . . 12
3.2.2. URDF File . . . 13
3.2.3. Movement of Model . . . 14

4. Pure PID control for autonomous robot **15**
4.1. Fundamental PID control theory . . . 15
4.2. PID controller design . . . 15
4.3. Process of PID control for autonomous robot . . . 16
4.4. PID control test . . . 17

5. Constrained Control of Mobile Manipulator **18**
5.1. Kinematic Equation Derivation . . . 18
5.2. Controller Design and Stability Analysis . . . 21

6. Comparison Result and Analysis **23**

7. Adaptive Neural Network Control of Simplified Manipulator **26**
7.1. Introduction of RBF NN-based Adaptive Control . . . 26
7.1.1. Feedback Linearization . . . 26
7.1.2. Linear Compensator and Error Dynamics . . . 27
7.2. Adaptive NN Controller on Simplified Manipulator . . . 27
7.2.1. Analysis of Stability . . . 30
7.3. Simulation . . . 32
7.4. Result . . . 33

8. Summary and Acknowledgement **35**
8.1. Summary of Project . . . 35
8.2. Acknowledgement . . . 35
8.3. Decleration . . . 35

List of Figures

1. CAD model of the autonomous robot . . . 12
2. Simulation results for XML files . . . 13
3. Simulation results for URDF files . . . 14
4. PID controller design . . . 16
5. Trajectory and error for tracking the first given trajectory . . . 17
6. Kinematics model of wheeled mobile robot. . . . 18
7. Structure chart of wheeled mobile robot. . . . 19
8. E velocity projection. . . . 19
9. Initial configuration of the autonomous robot . . . 23
10. True trajectory of the end-effecotr . . . 24
11. The projection of the trajectory . . . 24
12. The joint variables versus time . . . 24
13. The (angular) velocities versus time . . . 25
14. Radial basis function network structure. . . . 26
15. Structure of the whole control system. . . . 32
16. Network configuration. . . . 33
17. Desired vs. real q_1. . . . 33
18. Desired vs. real q_2. . . . 34

1. Literature Review

1.1. Introduction

The definition of an autonomous robot is a robot that has a variety of necessary sensors and controllers in its body and can perform tasks on its own with a high degree of autonomy when there is no external human information input and control during operation [1]. The characteristic of autonomy refers to the ability to operate normally in a real-world environment without any form of external control. For example, after obtaining information from the environment, it can exhibit various behaviors to adapt to environmental changes.

Autonomous systems created by humans often start from the perspective of biology and solving problems in real life. For example, unmanned aerial vehicles (UAVs) and micro aerial vehicles (MAVs) for aerial surveillance, reconnaissance, and inspection [2], autonomous robots for Monitoring Invasive Fish [3], a highly-maneuverable demining autonomous robot [4] and so on. All these autonomous systems can perform some tasks planned by humans in accordance with regulations. But as the complexity of the system design increases, robots may also perform unexpected behaviors in some emergency or unexpected situations. Autonomous robots at the current stage are mainly composed of vision systems, decision-making systems, underlying control systems, and communication systems.

Take autonomous vehicles as an example. According to the vision system and multiple sensors with different functions like millimeter-wave radar, fusion lidar, and camera, the system can obtain environmental information in the actual driving scene. Then the system will transmit the obtained information to the decision-making program. After getting information from the upper layer, reasonable decisions are made on the path planning and action planning of the autonomous vehicle and these commands will be sent to the underlying control system. Next step, when the underlying control system receives the command through the serial port, controllers in the system will control the vehicle's motor, steering click, throttle opening, and even lights.

1.2. Autonomous Robot Design

The development of autonomous robots is closely related to the development of computers, because the robot's architecture is based on the computer architecture. The foundation of the world's first general-purpose electronic computer is a digital electronic computer proposed by John von Neumann in a paper published in 1954 which is based on von Neumann architecture or the Princeton architecture [5]. The concepts of algorithms and formal procedures for solving linear and quadratic equations were first proposed here. But the original structure can only receive numbers or other information generated by human behavior and output "presumably only numerical information".

With the development of computers and technology, this structure has been continuously improved, which can accept various types of input information and can also carry out the non-numerical transmission of information. Proprioceptive sensors and exteroceptive sensors both play an important role in the process of realizing this structure [6].

Exteroceptive sensors are mainly used to obtain information about the external environment, such as pressure, hearing, and distance sensors. The proprioceptive sensors are mainly used to obtain internal state information, such as temperature, angle, and position sensors [7]. External effectors like motors are basically only used in the design of robotic arms or mobile robots. However, the information sensed from the outside can allow robots to react and change in response to the environment, which is the key to the success of autonomous machine design robots.

The proprioceptive sensors are often used together with internal effectors to maintain the dynamic balance of the controlled variables in a stable state through their respective feedback control. After the paper [8] was published, people regarded "*the property of the system to preserve its steady state byslow feedback control became known as homeostasis*" as one of the basic principles in autonomous machine design.

As mentioned in the paper [9] and [10], robots have three important properties: Situatedness, Embodiment, and Emergence. They correspond to the position of the robot in the environment, the feedback obtained after the robot itself interacts with the environment, and the interaction with the environment, respectively. These three properties are as important as the computing power of the controller and can allow people to draw inspiration from biology in the design of robots.

1.3. Virtual Robot Simulation

Virtual robot simulation refers to the technology of simulating the actual robot system through a computer. The geometry of the robot is generated on the simulation platform mainly through Computer Graphics Technology, Robotics Theory, and other knowledge. Then let it displays in two or three dimensions. The dynamic change process of the robot body and the working environment can be determined through simulation experiments. The design of the robot structure and algorithm can also be evaluated through the results at the same time.

However, there must exist differences between the actual robot and the simulated robot. Therefore, the significance of simulation is to provide an evaluation environment or object with good consistency and controllable uncertain factors. The main simulation factors mainly include the following three points: the kinematics and dynamics of the robot, the working environment of the robot, and the information returned by various sensors.

The simulation process before running and debugging on the actual platform is a very important step. For small robots, the efficiency of debugging programs directly through physical experiments may indeed be very high. However, for robots with higher complexity, if physical experiments are carried out first, it may cause software development to lag and fall into the process of constantly reorganizing the robot and debugging, which means the time and economic costs will be relatively high. Generally speaking, simulation can not only provide a stable debugging environment but also a low-cost learning environment that can maximize efficiency when robots completing complex tasks and provide an effective reference basis for manufacturing robots.

The development of simulation systems commonly used abroad began in 1982, and there are already many mature CAD software packages, such as DELMIA, ADAMS, ROBCAD, Matlab Simulink, Webots, Microsoft Robotics Studio, Ros, Gazebo, Utility 3D, and so on [11].

In order to judge the quality of each simulation software, it is necessary to formulate a unified standard. Mainly divided into the following four directions:

1. *Fidelity*: Mainly divided into physical fidelity and functional fidelity. Physical fidelity refers to the degree to which the appearance, sound, and feel of the physical environment are close to the real operating environment. Simulation software with high physical fidelity even includes a high degree of integration between the robot and the environment, and can also render high-resolution textures and materials. Functional fidelity refers to the degree to which the behavior of the robot in the simulation approaches the operating environment and equipment response of the real robot to perform tasks. Hope to get actions and results similar to actual experiments.
2. *Scalability*: Indicates the ability of the simulation software to use multiple applications.
3. *Development simplicity*: Including the complexity of the environment and robot configuration,

the types of programming languages supported, the versatility of communication interface configuration, etc.

4. *Cost*: Good simulation software hopes can let the time consumed during installation and use as small as possible.

There are many existing 2D and 3D simulation environments in OpenAI's gym package, including mobile manipulators, inverted pendulum cartpole, continuous inverted pendulums, biped robots, and other environments. Users can directly call for training in the python environment.

1.4. Constrained Control

Constraint systems include holonomic constraint systems and nonholonomic constraint systems. The holonomic constraint system equation does not include the derivative of coordinates for time, or the differential term in the constraint equation can be integrated into a finite form. The equation of a nonholonomic constraint system contains the differential of the coordinate that determines the system's position, and it cannot be directly integrated into an equation without coordinate differential.

Typical systems subject to nonholonomic constraints include vehicles, mobile robots, underwater robots, front-drive robots, and motion-restricted robots. Simply speaking, the aircraft is constrained in all six degrees of freedom, which is a holonomic restraint system. However, like a car, it cannot be controlled in a specific direction or degree of freedom and can only follow the plane of motion. This is a nonholonomic constraint system.

So wheeled mobile manipulator is a typical non-holonomic control system. The problem of controlling the response of a given system to track the desired trajectory is one of the most critical problems in both control theory and practice[12]. Its trajectory tracking control problem has always been one of the hot topics studied by scholars at home and abroad. The control structure allows feedback of displacements, velocities, and contact forces[13].

Trajectory tracking is one of the core functions of motion control, and the goal of its control is to accurately track the designed reference trajectory. In the actual control system, it is difficult to accurately measure due to the uncertainty of the internal model parameters of the wheeled mobile manipulator, coupled with the unknown disturbance of the external environment, nonlinearity, and friction of the wheeled mobile manipulator itself. As a result, the trajectory tracking controller of the wheeled mobile manipulator designed has low accuracy and poor stability, and it is difficult to meet the requirements of high precision and high reliability in the industry. The constraint control refers to the control of only partial degrees of freedom, or with bounded input in any control method like PD control, PID control and adaptive control.

PID controlled is popular by simple principle, good adaptability, easy to use, in industrial production[14]. PID controller is widely used in the process, and it is one of the earliest developed control methods. For the control of flexible manipulator, according to its characteristics, it is generally used PD controller. Sun[15] used fuzzy control combined with neural network. The end vibration suppression and joint trajectory tracking of flexible manipulator are realized.

According to different design techniques, adaptive control is divided into three categories, self-adaptive control, based on adaptive control and linear perturbation adaptive control. Three kinds of adaptive control are adaptive. The controller controls its operation. Adaptive controller can be used in uncertain and local environments. Human intelligence components, feel and identify the environment in operation, make decisions automatically, and execute decisions. In [16], adaptive control was proposed for trajectory or force control of mobile manipulators subjected to holonomic

and nonholonomic constraints with unknown inertia parameters, which ensures the movement of the system to asymptotically converge to the desired trajectory and force.

In response to the control goal, scholars have proposed different control strategies. Dong Xu et al.[17] used the neural network sliding mode control method to realize the trajectory tracking control of the wheeled mobile manipulator. Among them, the neural network control method is used to identify the uncertainties of unmodeled dynamics in the dynamic system. The sliding mode control method is used to suppress interference and ensure the stability of the system. Shihabudheen KV et al.[18] proposed the stability control and trajectory tracking control of a two-link wheeled mobile manipulator. The dynamic equation of the wheeled mobile manipulator was established by the Lagrangian method, and the wheeled mobile was designed by the sliding mode method. The controller of the robotic arm verifies the stability of the designed controller through simulation results. Kimitoshi Yamazaki et al.[19] described a novel method for trajectory tracking control of a wheeled mobile manipulator. A large number of mechanical arms are installed on the wheeled robot to ensure that it can run safely and without collision along the designed trajectory. The position and posture of the robot are calculated by the method of Jacobian matrix. A simple collision avoidance criterion is also adopted to ensure the stability of the system.

2. Lagrange-Euler Equations Derivation

This part will derive the Lagrange-Euler equations for the whole autonomous robot and the individual parts of the robotic arm. Tabla 1 defines the parameters involved in the autonomous robot system. This table will be used many times in the following.

Table 1: Meaning of parameters in system.

A	Connection point between first link and car
C	Centroid of car
D	Distance between A and C
θ	Angle between car and x-axis direction
θ_1, θ_2, θ_3	Rotation angle of each joint
L_0, L_1, L_2	Distance from connection point to centroid
l_0, l_1, l_2	Length of each link
m_0, m_1, m_2, m_3	Weight of car and each link
J_0, J_1, J_2, J_3	Moment of inertia

2.1. Robot Arm

This part only considers the robotic arm part. Moment of inertia is represented by an inertia matrix and its format is as follows:

$$I = \begin{bmatrix} Ixx_n & Ixy_n & Ixz_n \\ Iyx_n & Iyy_n & Iyz_n \\ Izx_n & Izy_n & Izz_n \end{bmatrix} \tag{1}$$

The position of each arm can be represented like this:

$$\begin{cases} x_1 = 0 \\ y_1 = 0 \\ z_3 = L_0 \end{cases}, \begin{cases} x_2 = L_1 c_1 c_2 \\ y_2 = L_1 s_1 c_2 \\ z_2 = l_0 - L_1 s2 \end{cases}, \begin{cases} x_3 = c_1 \left(L_2 c_{23} + l_1 c_2\right) \\ y_3 = s_1 \left(L_2 c_{23} + l_1 c_2\right) \\ z_3 = l_0 - l_1 s2 - L_2 s_{23} \end{cases} \tag{2}$$

Then we can get the equation like this:

$$H = \sum_{i=1}^{3} \left(m_i J_L^{(i)T} J_L^{(i)} + J_A^{(i)T} I_i J_A^{(i)} \right) = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{23} & H_{33} \end{bmatrix} \tag{3}$$

$$\begin{aligned} H_{11} =& Izz_1 + Izz_2 + Izz_3 + m_3 \left(L_2 c_{23} + l_1 c_2\right)^2 + m_2 L_1^2 c_2^2 \\ H_{12} =& H_{21} = \left(Iyz_2 + Iyz_3\right) c_1 - \left(Ixz_2 + Ixz_3\right) s_1 \\ H_{13} =& H_{31} = Iyz_3 c_1 - Ixz_3 s_1 \\ H_{22} =& Ixx_2 + Ixx_3 + m_2 L_1^2 + m_3 L_2^2 + m_3 l_1^2 + \left(Iyy_2 + Iyy_3 - Ixx_2 - Ixx_3\right) c_1^2 - \\ & 2\left(Ixy_2 + Ixy_3\right) s_1 c_1 + 2 m_3 l_1 L_2 c_3 \\ H_{23} =& H_{32} = Ixx_3 + m_3 L_2^2 + \left(Iyy_3 - Ixx_3\right) c_1^2 - 2 Ixy_3 s_1 c_1 + m_3 l_1 L_2 c_3 \\ H_{33} =& m_3 L_2^2 + Iyy_3 c_1^2 + Ixx_3 s_1^2 - 2 Ixy_3 s_1 c_1 \end{aligned} \tag{4}$$

After we get inertia matrix, we can get coriolis and centrifugal forces matrix which combined $C(q,\dot{q})$ and $\dot{q}$ as shown in the following process:

$$h_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial H_{jk}}{\partial q_i} \tag{5}$$

when i=1:

$$\begin{aligned}
h_{111} =& 0 \\
h_{112} =& -2m_3\left(L_2c_{23}+l_1c_2\right)\left(L_2s_{23}+l_1s_2\right) - 2m_2L_1^2s_2c_2 + \frac{1}{2}\left(Iyz_2+Iyz_3\right)s_1 + \frac{1}{2}\left(Ixz_2+Ixz_3\right)c_1 \\
h_{113} =& -2m_3\left(L_2c_{23}+l_1c_2\right)L_2s_{23} + \frac{1}{2}\left(Iyz_3s_1+Ixz_3c_1\right) \\
h_{121} =& -\frac{1}{2}\left[\left(Iyz_2+Iyz_3\right)s_1 + \frac{1}{2}\left(Ixz_2+Ixz_3\right)c_1\right] \\
h_{122} =& \left(Iyy_2+Iyy_3-Ixx_2-Ixx_3\right)s_1c_1 + \left(Ixy_2+Ixy_3\right)\left(2c_1^2-1\right) \\
h_{123} =& \left(Iyy_3-Ixx_3\right)s_1c_1 + Ixy_3\left(2c_1^2-1\right) \\
h_{131} =& -\frac{1}{2}\left(Iyz_3s_1+Ixz_3c_1\right) \\
h_{132} =& h_{123} \\
h_{133} =& Iyy_3s_1c_1 - Ixx_3s_1c_1 + Ixy_3\left(2c_1^2-1\right)
\end{aligned} \tag{6}$$

when i=2:

$$\begin{aligned}
h_{211} =& m_3\left(L_2c_{23}+l_1c_2\right)\left(L_2s_{23}+l_1s_2\right) + m_2L_1^2s_2c_2 - \left(Iyz_2+Iyz_3\right)s_1 - \left(Ixz_2+Ixz_3\right)c_1 \\
h_{221} =& -2\left(Iyy_2+Iyy_3-Ixx_2-Ixx_3\right)s_1c_1 - \left(Ixy_2+Ixy_3\right)\left(2c_1^2-1\right) \\
h_{223} =& -2m_3l_1L_2s_3 \\
h_{231} =& -2\left(Iyy_3-Ixx_3\right)s_1c_1 - 2Ixy_3\left(2c_1^2-1\right) \\
h_{233} =& -m_3l_1L_2s_3 \\
h_{212} =& h_{213} = h_{222} = h_{232} = 0
\end{aligned} \tag{7}$$

when i=3:

$$\begin{aligned}
h_{311} =& m_3\left(L_2c_{23}+l_1c_2\right)L_2s_{23} - \left(Iyz_3s_1+Ixz_3c_1\right) \\
h_{321} =& -2\left(Iyy_3-Ixx_3\right)s_1c_1 - 2Ixy_3\left(2c_1^2-1\right) \\
h_{322} =& m_3l_1L_2s_3 \\
h_{323} =& -\frac{1}{2}m_3l_1L_2s_3 \\
h_{331} =& Ixx_3s_1c_1 - 2Iyy_3s_1c_1 - 2Ixy_3\left(2c_1^2-1\right) \\
h_{332} =& \frac{1}{2}m_3l_1L_2s_3 \\
h_{312} =& h_{313} = h_{333} = 0
\end{aligned} \tag{8}$$

If use another formation like $H(q)\ddot{q}+C\left(q,\dot{q}\right)\dot{q}+G\left(q\right)=\tau$, we have to use the equation like this:

$$C_{ijk}=\frac{\partial H_{kj}}{\partial q_i}-\frac{1}{2}\frac{\partial H_{ij}}{\partial q_k} \tag{9}$$

when i=1:

$$\begin{aligned}
C_{111}=&0\\
C_{112}=&m_3\left(L_2c_{23}+l_1c_2\right)\left(L_2s_{23}+l_1s_2\right)+m_2L_1^2s_2c_2-\left(Iyz_2+Iyz_3\right)s_1-\left(Ixz_2+Ixz_3\right)c_1\\
C_{113}=&-\left(Iyz_3s_1+Ixz_3c_1\right)+m_3\left(L_2c_{23}+l_1c_2\right)L_2s_{23}\\
C_{121}=&-\frac{1}{2}\left[\left(Iyz_2+Iyz_3\right)s_1+\frac{1}{2}\left(Ixz_2+Ixz_3\right)c_1\right]\\
C_{122}=&-2\left(Iyy_2+Iyy_3-Ixx_2-Ixx_3\right)s_1c_1-2\left(Ixy_2+Ixy_3\right)\left(2c_1^2-1\right)\\
C_{123}=&-2\left(Iyy_3-Ixx_3\right)s_1c_1-2Ixy_3\left(2c_1^2-1\right)\\
C_{131}=&-\frac{1}{2}\left(Iyz_3s_1+Ixz_3c_1\right)\\
C_{132}=&C_{123}\\
C_{133}=&-2Iyy_3s_1c_1+2Ixx_3s_1c_1-2Ixy_3\left(2c_1^2-1\right)
\end{aligned} \tag{10}$$

when i=2:

$$\begin{aligned}
C_{211}=&-2m_3\left(L_2c_{23}+l_1c_2\right)\left(L_2s_{23}+l_1s_2\right)-2m_2L_1^2s_2c_2+\frac{1}{2}\left(Iyz_2+Iyz_3\right)s_1+\frac{1}{2}\left(Ixz_2+Ixz_3\right)c_1\\
C_{221}=&\left(Iyy_2+Iyy_3-Ixx_2-Ixx_3\right)s_1c_1+\frac{1}{2}\left(Ixy_2+Ixy_3\right)\left(2c_1^2-1\right)\\
C_{223}=&m_3l_1L_2s_3\\
C_{231}=&\left(Iyy_3-Ixx_3\right)s_1c_1+Ixy_3\left(2c_1^2-1\right)\\
C_{233}=&\frac{1}{2}m_3l_1L_2s_3\\
C_{212}=&C_{213}=C_{222}=C_{232}=0
\end{aligned} \tag{11}$$

when i=3:

$$\begin{aligned}
C_{311}=&-2m_3\left(L_2c_{23}+l_1c_2\right)L_2s_{23}+\frac{1}{2}\left(Iyz_3s_1+Ixz_3c_1\right)\\
C_{321}=&\left(Iyy_3-Ixx_3\right)s_1c_1+Ixy_3\left(2c_1^2-1\right)\\
C_{322}=&-2m_3l_1L_2s_3\\
C_{323}=&-\frac{1}{2}m_3l_1L_2s_3\\
h_{331}=&Iyy_3s_1c_1-Ixx_3s_1c_1+Ixy_3\left(2c_1^2-1\right)\\
h_{332}=&-m_3l_1L_2s_3\\
C_{312}=&C_{313}=C_{333}=0
\end{aligned} \tag{12}$$

Each item in $C\left(q,\dot{q}\right)$ can be represented by $C_{kj}=\sum_{i=1}^{n}C_{ijk}\dot{q}_i$. Then gravity forces matrix can be

derived like this:

$$G_i = -\sum_{j=1}^{n} m_j g^T J_{Li}^{(i)} \tag{13}$$

$$G_1 = -m_1 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - m_2 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} -L_1 s_1 c_2 \\ L_1 c_1 c_2 \\ 0 \end{bmatrix} - m_3 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} -s_1 \left(L_2 c_{23} + l_1 c_2\right) \\ c_1 \left(L_2 c_{23} + l_1 c_2\right) \\ 0 \end{bmatrix} = 0 \tag{14}$$

$$\begin{aligned} G_2 =& - m_1 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - m_2 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} -L_1 c_1 s_2 \\ -L_1 s_1 s_2 \\ -L_1 c_2 \end{bmatrix} - m_3 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} -c_1 \left(L_2 c_{23} + l_1 c_2\right) \\ -s_1 \left(L_2 c_{23} + l_1 c_2\right) \\ -l_1 c_2 - L_2 c_{23} \end{bmatrix} \\ =& - m_2 g L_1 c_2 - m_3 g \left(l_1 c_2 + L_2 c_{23}\right) \end{aligned} \tag{15}$$

$$G_3 = -m_1 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - m_2 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} - m_3 \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}^T \begin{bmatrix} -L_2 c_1 s_{23} \\ -L_2 s_1 s_{23} \\ -L_2 c_{23} \end{bmatrix} = -m_3 g L_2 c_{23} \tag{16}$$

Combining these formulas and items together, we can get the equation of the robotic arm. These formulas can be simplified. If the manipulator is an object with uniform mass, the product of inertia of the link is zero.

2.2. Autonomous Robot

To get kinetic energy of the whole system, we have to know the expression of position and velocity of each part.

The positional relationship between A and C is as follows:

$$\begin{cases} x_A = x_C - D cos\theta \\ y_A = y_C - D sin\theta \end{cases} \tag{17}$$

The coordinates of the center of mass of each link are expressed as follows:

$$\begin{cases} x_1 = x_A \\ y_1 = y_A \\ z_3 = L_0 \end{cases} \tag{18}$$

$$\begin{cases} x_2 = x_1 + L_1 \cdot \cos\left(\theta + \theta_1\right) \cdot \cos\theta_2 \\ y_2 = y_1 + L_1 \cdot \sin\left(\theta + \theta_1\right) \cdot \cos\theta_2 \\ z_2 = l_0 - L_1 \cdot \sin\theta_2 \end{cases} \tag{19}$$

$$\begin{cases} x_3 = x_1 + \cos\left(\theta + \theta_1\right) \cdot \left[L_2 \cos\left(\theta_2 + \theta_3\right) + l_1 \cos\theta_2\right] \\ y_3 = y_1 + \sin\left(\theta + \theta_1\right) \cdot \left[L_2 \cos\left(\theta_2 + \theta_3\right) + l_1 \cos\theta_2\right] \\ z_3 = l_0 - 4\sin\theta_2 - L_2 \sin\left(\theta_2 + \theta_3\right) \end{cases} \tag{20}$$

We can get the expression of velocity according to the position. And then we can express kinetic

energy of the whole system like this:

$$\begin{aligned} T = \frac{1}{2}\sum_{i=1}^{6}\left(m_i v^2 + J\omega_i^2\right) =& \frac{1}{2}m_0\left(\dot{x}_C^2 + y_C^2\right) + \frac{1}{2}J_0\dot{\theta}^2 + \frac{1}{2}m_1\left(\dot{x}_1^2 + \dot{y}_1^2\right) + \frac{1}{2}J_1\left(\dot{\theta}+\dot{\theta}_1\right)^2 \\ &+ \frac{1}{2}m_2\left(\dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2\right) + \frac{1}{2}J_2\left[\left(\dot{\theta}+\dot{\theta}_1\right)^2 + \dot{\theta}_2^2\right] \\ &+ \frac{1}{2}m_3\left(\dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2\right) + \frac{1}{2}J_3\left[\left(\dot{\theta}+\dot{\theta}_1\right)^2 + \dot{\theta}_2^2 + \dot{\theta}_3^2\right] \end{aligned} \tag{21}$$

Lagrange-Euler equation is $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i$. To distinguish from the equation of the robotic arm, different parameter representations are replaced. We can write the motion equations in matrix form as follow:

$$D\left(q\right)\ddot{q} + C\left(q,\dot{q}\right)\dot{q} + G\left(q\right) = \tau \tag{22}$$

where q means joint variable vector, $D\left(q\right)$ means inertia matrix, $C\left(q,\dot{q}\right)$ is coriolis and centrifugal forces matrix, $G\left(q\right)$ is gravity forces matrix and τ is the vector of input. Since the car is moving in a plane, the gravitational potential energy does not change. So its gravity forces matrix is the same as above $G(q)$. After calculating, we can get the expressions of each matrix as follows:

$$D\left(q\right) = \begin{bmatrix} D_{11} & 0 & D_{13} & D_{14} & D_{15} & D_{16} \\ 0 & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} \\ D_{31} & D_{32} & D_{33} & D_{34} & 0 & 0 \\ D_{41} & D_{42} & D_{43} & D_{44} & 0 & 0 \\ D_{51} & D_{52} & 0 & 0 & D_{55} & D_{56} \\ D_{61} & D_{62} & 0 & 0 & D_{65} & D_{66} \end{bmatrix} \tag{23}$$

$$\begin{aligned} &D_{11} = D_{22} = m_0 + m_1 + m_2 + m_3 \\ &D_{13} = D_{31} = -2m_3\left[\sin\left(\theta+\theta_1\right)\left(L_2\cos\left(\theta_2+\theta_3\right) + l_1\cos\theta_2\right)\right] - 2m_2L_1\cos\theta_2\sin\left(\theta+\theta_1\right) - 2m_0D\sin\theta \\ &D_{14} = D_{41} = -2m_3\left[\sin\left(\theta+\theta_1\right)\left(L_2\cos\left(\theta_2+\theta_3\right) + l_1\cos\theta_2\right)\right] - 2m_2L_1\cos\theta_2\sin\left(\theta+\theta_1\right) \\ &D_{15} = D_{51} = -2m_3\cos\left(\theta+\theta_1\right)\left(L_2\sin\left(\theta_2+\theta_3\right) + l_1\sin\theta_2\right) - 2m_2L_1\sin\theta_2\cos\left(\theta+\theta_1\right) \\ &D_{16} = D_{61} = -2m_3\cos\left(\theta+\theta_1\right)L_2\sin\left(\theta_2+\theta_3\right) \\ &D_{23} = D_{32} = 2m_0D\cos\theta + 2m_2L_1\cos\theta_2\cos\left(\theta+\theta_1\right) + 2m_3\cos\left(\theta+\theta_1\right)\left[L_2\cos\left(\theta_2+\theta_3\right) + h\cos\theta_2\right] \\ &D_{24} = D_{42} = 2m_2L_1\cos\theta_2\cos\left(\theta+\theta_1\right) + 2m_3\cos\left(\theta+\dot{\theta}\right)\left[L_2\cos\left(\theta_2+\theta_3\right) + l_1\cos\theta_2\right] \\ &D_{25} = D_{52} = -2m_2L_1\sin\theta_2\sin\left(\theta+\theta_1\right) - 2m_3\sin\left(\theta+\theta_1\right)\left[L_2\sin\left(\theta_2+\theta_3\right) + l_1\sin\theta_2\right] \\ &D_{26} = D_{62} = -2m_3\sin\left(\theta+\theta_1\right)L_2\sin\left(\theta_2+\theta_3\right) \\ &D_{33} = J_0 + J_1 + J_2 + J_3 + m_0D^2 + m_2L_1^2\cos^2\theta_2 + m_3\left[L_2\cos\left(\theta_2+\theta_3\right) + l_1\cos\theta_2\right]^2 \\ &D_{34} = D_{43} = 2\left(J_1 + J_2 + J_3\right) + 2m_2L_1^2\cos^2\theta_2 + 2m_3\left[L_2\cos\left(\theta_2+\theta_3\right) + l_1\cos\theta_2\right]^2 \\ &D_{44} = J_1 + J_2 + J_3 + m_0D^2 + m_2L_1^2\cos^2\theta_2 + m_3\left[L_2\cos\left(\theta_2+\theta_3\right) + l_1\cos\theta_2\right]^2 \\ &D_{55} = J_2 + J_3 + m_2L_1^2 + m_3l_1^2 + m_3L_2^2 + 2m_3l_1L_2\cos\theta_3 \\ &D_{56} = 2m_3L_2^2 + 2m_3l_1L_2\cos\theta_3 \\ &D_{66} = J_3 + m_3L_2^2 \end{aligned} \tag{24}$$

After getting the matrix $D(q)$, follow the same process to calculate the matrix $C(q,\dot{q})$.

$$C\left(q,\dot{q}\right)=\begin{bmatrix} 0 & 0 & C_{13} & C_{14} & C_{15} & C_{16} \\ 0 & 0 & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{54} & C_{65} & C_{66} \end{bmatrix} \tag{25}$$

$$\begin{aligned}
C_{13} =& 2C_{31} = (2m_3 sin(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+2L_1 m_2 sin(\theta+\theta_1)sin\theta_2)\dot{\theta_2} \\
&-(2m_3 cos(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2L_1 m_2 cos(\theta+\theta_1)cos\theta_2)\dot{\theta_1} \\
&-(2m_3 cos(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2m_0 D cos\theta+2L_1 m_2 cos(\theta+\theta_1)cos\theta_2)\dot{\theta} \\
&+2L_2 m_3 sin(\theta+\theta_1)sin(\theta_2+\theta_3)\dot{\theta_3} \\
C_{14} =& C_{41} = 2(2m_3 sin(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+2L_1 m_2 sin(\theta+\theta_1)sin\theta_2)\dot{\theta_2}- \\
&(2m_3 cos(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2L_1 m_2 cos(\theta+\theta_1)cos\theta_2)(\dot{\theta_1}+\dot{\theta}) \\
&+2L_2 m_3 sin(\theta+\theta_1)sin(\theta_2+\theta_3)\dot{\theta_3} \\
C_{15} =& 2C_{51} = (2m_3 sin(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+2L_1 m_2 sin(\theta+\theta_1)sin\theta_2)(\dot{\theta_1}+\dot{\theta}) \\
&-(2m_3 cos(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2L_1 m_2 cos(\theta+\theta_1)cos\theta_2)\dot{\theta_2} \\
&-2L_2 m_3 cos(\theta+\theta_1)cos(\theta_2+\theta_3)\dot{\theta_3} \\
C_{16} =& 2C_{61} = 2L_2 m_3 sin(\theta+\theta_1)sin(\theta_2+\theta_3)(\dot{\theta_1}+\dot{\theta})-2L_2 m_3 cos(\theta+\theta_1)cos(\theta_2+\theta_3)(\dot{\theta_2}+\dot{\theta_3}) \\
C_{23} =& 2C_{32} = -(2m_3 sin(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2L_1 m_2 sin(\theta+\theta_1)cos\theta_2)\dot{\theta_1} \\
&-(2m_3 sin(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2Dm_0 sin\theta+2L_1 m_2 sin(\theta+\theta_1)cos\theta_2)\dot{\theta} \\
&-(2m_3 cos(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+2L_1 m_2 cos(\theta+\theta_1)sin\theta_2)\dot{\theta_2} \\
&-2L_2 m_3 cos(\theta+\theta_1)sin(\theta_2+\theta_3)\dot{\theta_3} \\
C_{24} =& 2C_{42} = -(2m_3 sin(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2L_1 m_2 sin(\theta+\theta_1)cos\theta_2)(\dot{\theta_1}+\dot{\theta}) \\
&-(2m_3 cos(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+2L_1 m_2 cos(\theta+\theta_1)sin\theta_2)\dot{\theta_2} \\
&-2L_2 m_3 cos(\theta+\theta_1)sin(\theta_2+\theta_3)\dot{\theta_3} \\
C_{25} =& 2C_{52} = -(2m_3 sin(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+2L_1 m_2 sin(\theta+\theta_1)cos\theta_2)\dot{\theta_2} \\
&-(2m_3 cos(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+2L_1 m_2 sin(\theta+\theta_1)cos\theta_2)(\dot{\theta_1}+\dot{\theta}) \\
&-2L_2 m_3 cos(\theta+\theta_1)sin(\theta_2+\theta_3)\dot{\theta_3} \\
C_{26} =& 2C_{62} = -2L_2 m_3 cos(\theta+\theta_1)sin(\theta_2+\theta_3)(\dot{\theta}+\dot{\theta_q})-2L_2 m_3 sin(\theta+\theta_1)cos(\theta_2+\theta_3)(\dot{\theta_2}+\dot{\theta_3})
\end{aligned} \tag{26}$$

$$\begin{aligned}
C_{33} =& (m_3 cos(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+m_0 D cos\theta+L_1 m_2 cos(\theta+\theta_1)cos\theta_2)\dot{x} \\
&+(m_3 sin(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+m_0 D sin\theta+L_1 m_2 sin(\theta+\theta_1)cos\theta_2))\dot{y} \\
&-(2m_2 cos\theta_2 sin\theta_2 L_1^2+2m_3(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)(L_2 sin(\theta_2+\theta_3)+l_1 sin(\theta_2)))\dot{\theta_2} \\
&-2L_2 m_3 sin(\theta_2+\theta_3)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)\dot{\theta_3} \\
C_{34} =& C_{43} = (cos(\theta+\theta_1)\dot{x}+sin(\theta+\theta_1)\dot{y})(l_1 m_3 cos\theta_2+L_2 m_3 cos(\theta_2+\theta_3)+L_1 m_2 cos\theta_2) \\
&-(4m_2 cos\theta_2 sin\theta_2 L_1^2+4m_3(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)\dot{\theta_2} \\
&-4L_2 m_3 sin(\theta_2+\theta_3)(L_2 cos(q2+q3)+l_1 cos\theta_2)\dot{\theta_3} \\
C_{35} =& C_{45} = (cos(\theta+\theta_1)\dot{y}-sin(\theta+\theta_1)\dot{x})(l_1 m_3 sin\theta_2+L_2 m_3 sin(\theta_2+\theta_3)+L_1 m_2 sin\theta_2) \\
C_{36} =& C_{46} = L_2 m_3 sin(\theta_2+\theta_3)(cos(\theta+\theta_1)\dot{y}-sin(\theta+\theta_1)\dot{x})
\end{aligned} \tag{27}$$

$$\begin{aligned}
C_{33} =& (m_3 cos(\theta+\theta_1)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)+m_0 D cos\theta+L_1 m_2 cos(\theta+\theta_1)cos\theta_2)\dot{x} \\
&+(m_3 sin(\theta+\theta_1)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)+m_0 D sin\theta+L_1 m_2 sin(\theta+\theta_1)cos\theta_2))\dot{y} \\
&-(2m_2 cos\theta_2 sin\theta_2 L_1^2+2m_3(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)(L_2 sin(\theta_2+\theta_3)+l_1 sin(\theta_2)))\dot{\theta_2} \\
&-2L_2 m_3 sin(\theta_2+\theta_3)(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)\dot{\theta_3} \\
C_{34} =& C_{43} = (cos(\theta+\theta_1)\dot{x}+sin(\theta+\theta_1)\dot{y})(l_1 m_3 cos\theta_2+L_2 m_3 cos(\theta_2+\theta_3)+L_1 m_2 cos\theta_2) \\
&-(4m_2 cos\theta_2 sin\theta_2 L_1^2+4m_3(L_2 cos(\theta_2+\theta_3)+l_1 cos\theta_2)(L_2 sin(\theta_2+\theta_3)+l_1 sin\theta_2)\dot{\theta_2} \\
&-4L_2 m_3 sin(\theta_2+\theta_3)(L_2 cos(q2+q3)+l_1 cos\theta_2)\dot{\theta_3} \\
C_{35} =& C_{45} = (cos(\theta+\theta_1)\dot{y}-sin(\theta+\theta_1)\dot{x})(l_1 m_3 sin\theta_2+L_2 m_3 sin(\theta_2+\theta_3)+L_1 m_2 sin\theta_2) \\
C_{36} =& C_{46} = L_2 m_3 sin(\theta_2+\theta_3)(cos(\theta+\theta_1)\dot{y}-sin(\theta+\theta_1)\dot{x})
\end{aligned} \tag{28}$$

$$
\begin{aligned}
C_{44} =&(cos(\theta+\theta_1)\dot{x}+sin(\theta+\theta_1)\dot{y})(l_1m_3cos\theta_2+L_2m_3cos(\theta_2+\theta_3)+L_1m_2cos\theta_2)\\
&-(2m_2cos\theta_2sin\theta_2L_1^2+2m_3(L_2cos(\theta_2+\theta_3)+l_1cos\theta_2)(L_2sin(\theta_2+\theta_3)+l_1sin\theta_2)\dot{\theta_2}\\
&-2L_2m_3sin(\theta_2+\theta_3)(L_2cos(q2+q3)+l_1cos\theta_2)\dot{\theta_3}
\end{aligned} \tag{29}
$$

$$
\begin{aligned}
C_{53} =&2C_{54}=(m_2cos\theta_2sin\theta_2L_1^2+m_3(L_2cos(\theta_2+\theta_3)+l_1cos\theta_2)(L_2sin(\theta_2+\theta_3)+l_1sin\theta_2))\dot{x}\\
&+(cos(\theta+\theta_1)\dot{y}-sin(\theta+\theta_1)\dot{x})(l_1m_3sin\theta_2+L_2m_3sin(\theta_2+\theta_3)+L_1m_2sin\theta_2)\\
&+(2m_2cos\theta_2sin\theta_2L_1^2+2m_3(L_2cos(\theta_2+\theta_3)+l_1cos\theta_2)(L_2sin(\theta_2+\theta_3)+l_1sin\theta_2))\dot{\theta_1}
\end{aligned} \tag{30}
$$

$$
\begin{aligned}
C_{55} =&(cos(\theta+\theta_1)\dot{x}+sin(\theta+\theta_1)\dot{y})(l_1m_3cos\theta_2+L_2m_3cos(\theta_2+\theta_3)+L_1m_2cos\theta_2)-2l_1L_2m_2sin\theta_3\dot{\theta_3}\\
C_{56} =&L_2m_3cos(\theta_2+\theta_3)[cos(\theta+\theta_1)\dot{x}+sin(\theta+\theta_1)\dot{y}]-2l_1L_2m_3sin\theta_3\dot{\theta_3}
\end{aligned}
$$

$$
\begin{aligned}
C_{63} =&L_2m_3sin(\theta_2+\theta_3)(cos(\theta+\theta_1)\dot{y}-sin(\theta+\theta_1)\dot{x}+2l_1cos\theta_2\dot{\theta_1}+l_1cos\theta_2\dot{x}\\
&+2L_2cos(\theta_2+\theta_3)\dot{\theta_1}+L_2cos(\theta_2+\theta_3)\dot{x})\\
C_{64} =&L_2m_3sin(\theta_2+\theta_3)(cos(\theta+\theta_1)\dot{y}-sin(\theta+\theta_1)\dot{x}+l_1cos\theta_2\dot{\theta_1}+2l_1cos\theta_2\dot{x}\\
&+L_2cos(\theta_2+\theta_3)\dot{\theta_1}+2L_2cos(\theta_2+\theta_3)\dot{x})\\
C_{65} =&L_2m_3cos(\theta_2+\theta_3)[cos(\theta+\theta_1)\dot{x}+sin(\theta+\theta_1)\dot{y}]-2l_1L_2m_3sin\theta_3\dot{\theta_3}+l_1L_2m_2sin\theta_3\dot{\theta_2}+l_1L_2m_3sin\theta_3\dot{\theta_3}\\
C_{66} =&L_2m_3cos(\theta_2+\theta_3)[cos(\theta+\theta_1)\dot{x}+sin(\theta+\theta_1)\dot{y}]+l_1L_2m_3sin\theta_3\dot{\theta_2}
\end{aligned} \tag{31}
$$

3. CAD Model and Model Import

3.1. CAD model of the autonomous robot

In this subsection we introduce the CAD model of the autonomous robot developed in SolidWorks.

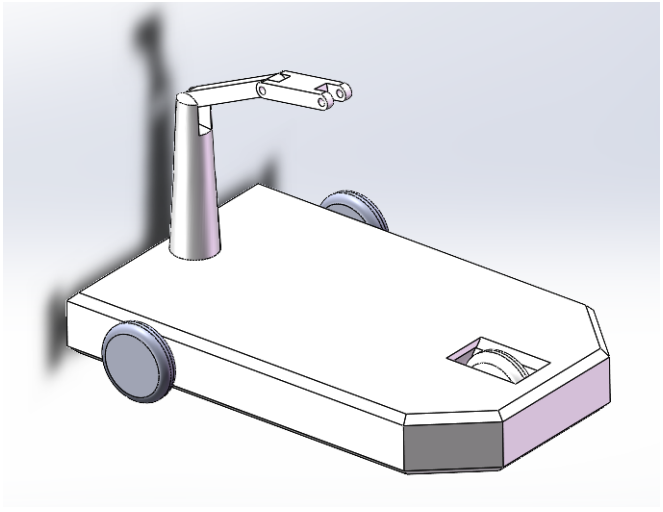


Figure 1: CAD model of the autonomous robot

The autonomous robot consists of two part, one is the autonomous car and one is the manipulator. The autonomous car contains three wheels and it is connected the manipulator by a revolute joint. The manipulator consists of three links which are connected by the revolute joints. The properties of the autonomous robot is as below.

	mass/kg	length/mm	$I_{xx}/(kg/m^2)$	$I_{yy}/(kg/m^2)$	$I_{zz}/(kg/m^2)$
Car	127.64	/	178.1	169.89	19.93
link1	2.77	40	5.71	5.19	0.53
link2	0.51	30	1.46	1.41	0.05
link3	0.54	20	1.67	1.66	0.00

3.2. Import Model to Matlab

This part mainly introduces how to import the robot model in SolidWorks into Matlab for experiments and the problems encountered.We tried two methods, including importing XML files and URDF files. The following is a detailed process introduction.

3.2.1. XML File

MatLab supports import XML Schema works with Simscape Multibody. Use the command “smimport” of Robotics System Toolbox to convert the XML file into the SLX file. At the same time, we can get a DATAfile file that stores the required parameters in Simulink. Run this file to see the car

model. Since each joint's motion limits were not set at the beginning, each joint will rotate under the action of gravity.

If we want to use the car model in the m file, we need to use the "importrobot" command to convert the car model to rigidbodyTree format for storage. The rigidBodyTree is a representation of the connectivity of rigid bodies with joints. After importing the car model, we can simulate its trajectory following. However, since the format of the entity visualization file of the XML file is STEP, which is different from the visualization file format required by the robot. Therefore, the dynamic image obtained only includes the coordinate relationship between each link which is shown as follow:

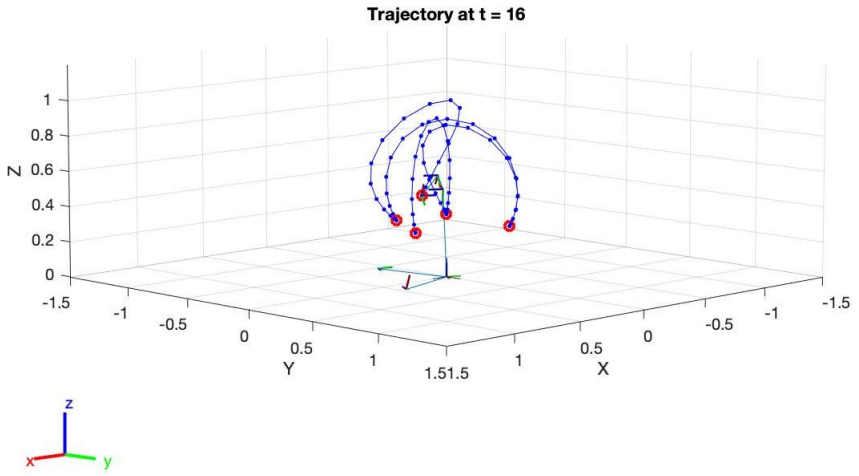


Figure 2: Simulation results for XML files

3.2.2. URDF File

After checking the references and website, we choose to change and use URDF to import. Now we can plot the car model with visualization files which is shown as follow.

However, due to the limitation of the joint angle in SolidWorks, the robot cannot move. Faced with this problem, we try to change the joint limits in SLX file and then convert it to the rigifbodyTree format. However, even though the visualization file in STL format has been defined and added in the SLX file, the car entity can only be seen when the Simulink file is run. After quoting the SLX file as rigidbodyTree format, it can still only display the relationship between the coordinate axes.

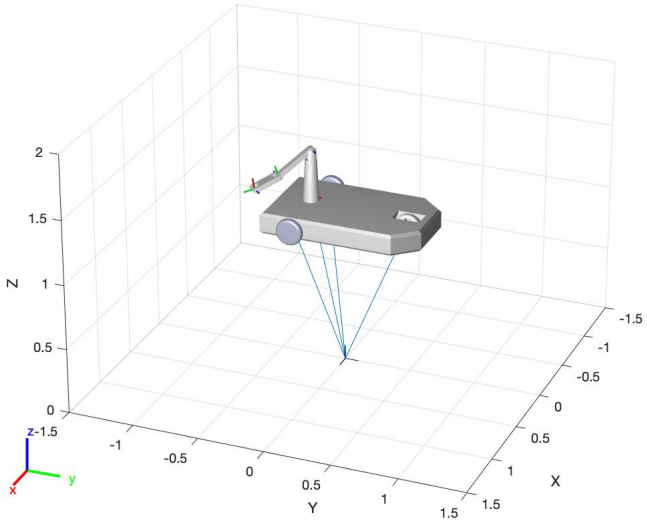


Figure 3: Simulation results for URDF files

3.2.3. Movement of Model

During Solidworks' modeling process, the first part added will be connected to the world coordinate by default to see the solid line connected to the above figure's zero points. The variable controlled after importing the model is the rotation angle of each joint. But this variable cannot reflect the displacement of the car. Therefore, if we want to realize the car's displacement and rotation, we can only simulate in Simulink. Because some joints such as 6-DOF, Planar Joint cannot be converted to rigidbodyTree format.

Another problem is that rigidbodyTree is mainly used for the simulation of robotic arms. Therefore, when the car's rotation angle is added, although the variable's weight is changed, the result will be that the trolley rotates greatly and frequently during the trajectory planning process.

4. Pure PID control for autonomous robot

In this section, a pure PiD control for the autonomous robot is designed to make the autonomous robot able to track any given desired trajectory.

4.1. Fundamental PID control theory

PID control consists of three control terms which are proportional part, integral part and derivative part. PID control continuously track an error $e(t)$ as the difference between the desired setpoint and a measured process variable and then adjust the input of the system according to the error. The mathematical form of PID control is as below:

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt}$$

where K_p, K_i and K_d , all non-negative, denote the coefficients for the proportional, integral, and derivative terms respectively.

The proportional component depends only on the difference between the set point and the process variable. This difference is referred to as the error term. The proportional gain determines the ratio of output response to the error signal.

The integral component sums the error term over time. The result is that even a small error term will cause the integral component to increase slowly. The integral response will continually increase over time unless the error is zero, so the effect is to drive the Steady-State error to zero. Steady-State error is the final difference between the process variable and set point.

The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. The Derivative Response is highly sensitive to noise in the process variable signal so a small sample interval is needed to make the system stable.

In this section, we consider the desired setpoint is a desired trajectory and the measured process variable is the position of the end-effector. We apply PID controller to control the the autonomous robot to reduce the error between the desired trajectory and the true trajectory.

4.2. PID controller design

In this subsection we discuss the PD controller design for autonomous robot system. In this project, we consider to control the autonomous robot by controlling the (angle) velocity of each joint. The input of the system is the desired trajectory, noted as θ_d. As described in the last subsection, we consider the distance between desired trajectory and the true position of the end-effector as the error e, which is the input of the PID controller. The output of the PID controller is the velocity of the end-effector which control the movement of the autonomous robot. The control process is as below and the detail of the process will be discussed in next subsetion.

Figure 4: PID controller design

4.3. Process of PID control for autonomous robot

Firstly we track the error of the system. In last section we model the autonomous robot with six joint variables, which are $x, y, \theta, \theta_1, \theta_2, \theta_3$. x, y, θ, describe the position and posture of the autonomous car and $\theta_1, \theta_2, \theta_3$ describe the state of the manipulator. These joint variables consists a joint variable vector $q = [x, y, \theta, \theta_1, \theta_2, \theta_3]^T$. For the autonomous robot, we can develop the transformation matrices as below:

$$
{}^1_0A(x) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^2_1A(y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^3_2A(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^4_3A(\theta_1) = \begin{bmatrix} cos\theta_1 & 0 & -sin\theta_1 & 0 \\ sin\theta_1 & 0 & cos\theta_1 & 0 \\ 0 & -1 & 0 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^5_4A(\theta_2) = \begin{bmatrix} cos\theta_2 & -sin\theta_2 & 0 & l_1cos\theta_2) \\ sin\theta_2 & cos\theta_2 & 0 & l_1sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^6_5A(\theta_3) = \begin{bmatrix} cos\theta_3 & -sin\theta_3 & 0 & l_2cos\theta_3 \\ sin\theta_3 & cos\theta_3 & 0 & l_2sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

With the transformation matrix it is easy to compute the end-effector position $X_{EE} = [x_{EE}, y_{EE}, z_{EE}]^T$ by the transformation matrix.

$$
{}^0_6T = {}^1_0\ A(x){}^2_1A(y){}^3_2A(\theta){}^4_3A(\theta_1){}^5_4A(\theta_2){}^6_5A(\theta_3)
$$

Comparing with the given trajectory at the same time step X_{EEref}, we compute the distance between the two point at each dimension as the error $e(t)$.

Second we discuss the PD control term. As we have computed the Jacobian matrix of the autonomous robot as below:

$$
J^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -sin(\theta+\theta_1)(L_2cos(\theta_2+\theta_3)+l_1cos_3) & cos(\theta+\theta_1)(L_2cos(\theta_2+\theta_3)+l_1cos_3) & 0 \\ -sin(\theta+\theta_1)(L_2cos(\theta_2+\theta_3)+l_1cos_3) & cos(\theta+\theta_1)(L_2cos(\theta_2+\theta_3)+l_1cos_3) & 0 \\ -cos(\theta+\theta_1)(L_2cos(\theta_2+\theta_3)+l_1cos_3) & -sin(\theta+\theta_1)(L_2cos(\theta_2+\theta_3)+l_1cos_3) & -L_2cos(\theta_2+\theta_3)-l_1cos_3 \\ -L_2cos(\theta+\theta_1)sin(\theta_2+\theta_3) & -L_2sin(\theta+\theta_1)sin(\theta_2+\theta_3) & -L_2cos(\theta_2+\theta_3) \end{bmatrix}
$$

only the velocity of end-effector is needed to compute the velocities of all the joints. Therefore we

use the PD controller to control the velocity of the end-effector. The coefficient of the PD controller is set as $K_p = 1.5, K_i = 0.001, K_d = 0.5$. The PD controller outputs velocity of the end-effector.

$$\dot{X}(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt}$$

Then we apply the inverse Jacobian matrix to compute the velocity of all the joints.

$$\dot{q}(t) = J^T \dot{X}(t)$$

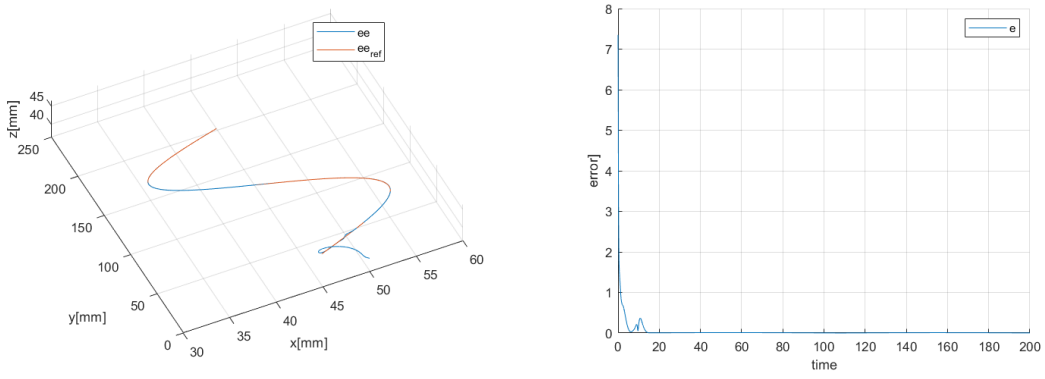
Taking $\dot{q}$ as the input of the autonomous robot system, it conducts the joint variable vector at next time step and so dose the position of the end-effector. Repeat the operation in first step so the error at next time step can be computed and taken as the input of PID controller for tracking the trajectory continuously.

4.4. PID control test

We test our PID controller to track a given trajectory to validate the rationality of the PID parameters setting. In the test, we set the sample interval as 0.001s, the running time as 200s and the initial joint variables as $[0, 0, 0, 0, 0, 0]$.We display the performance of the PID control with the trajectory and the error in the simulation.

The given trajectory to track is as below:

$$\begin{cases} x = 10sin(0.01\pi t) + 45 \\ y = 2 + t \\ z = 45 * cos(0.001\pi t) \end{cases} \qquad 0 \leq t \leq 200$$



(a) Trajectory (b) Error

Figure 5: Trajectory and error for tracking the first given trajectory

From the test result we can see the robust property of the PID control. Even within drastic change in desired trajectory, the autonomous robot still has stable performance.

5. Constrained Control of Mobile Manipulator

In this section, we first do the kinematic equation derivation of the mobile manipulator. Then based on the kinematic equation we design a control law to control the trajectory of the end-effector of the mobile manipulator based on the joint velocities.

5.1. Kinematic Equation Derivation

For this project, we use the differential drive model. In the differential drive model, the difference between two driving wheels decides the direction of motion of the car body. The average velocity of two wheels decides the velocity of the car body.

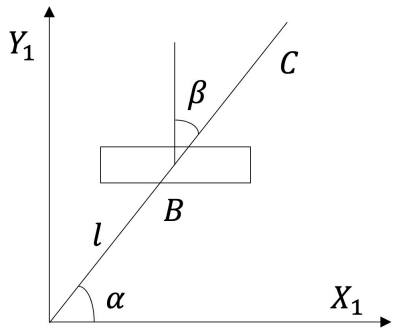


Figure 6: Kinematics model of wheeled mobile robot.

We assume that the robot is consist of rigid body and rigid wheels and assume that the robot moves on a rigid plane. Below is the kinematic model of the car. xoy is the global Cartesian coordinate of the motion plane of car. C is the center of robot, which in our model is the center of wheel line. Therefore we can describe the full configuration of car body with a set of vector (x, y, θ).

Due to mechanical constraints, the rolling wheels of the mobile trolley will not move in a direction perpendicular to the plane of the wheels. That is, for the two driving wheels, the center point O cannot individually move along axis y_I. Therefore, the car is constrained by the nonholonomic constraint.

The figure below shows the simplified structure graph of the car. For the differential driven car, we set the key point B as the origin of the car body. The position of B in moving coordinate system (x_l, y_l) is described by length $CB = l$ and angle α. The wheel plane direction is described by a constant angle β with respect to the direction of CB. The rotation angle of wheel with respect to its horizontal axis is denoted by $\varphi(t)$, and the radius of wheel is r. In all, the state of wheel is described by four constants: $\alpha, \beta, l$and r, and the state of motion is described by the angle $\varphi(t)$. Assume that the wheel is pure rolling without sliding on the plane.

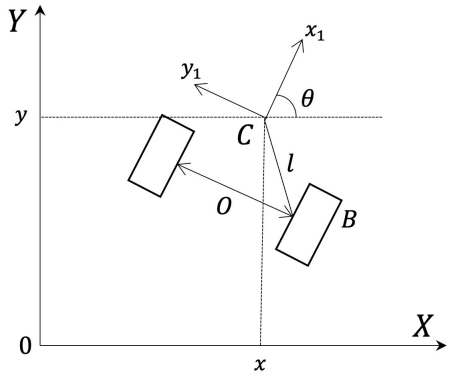


Figure 7: Structure chart of wheeled mobile robot.

According to the basic velocity transformation law of rigid body, we can easily get the constraint condition of the differential driven car:

$$r\dot{\varphi} = (\dot{x}\cos\theta + \dot{y}\sin\theta)\sin(\alpha+\beta) - (-\dot{x}\sin\theta + \dot{y}\cos\theta)\cos(\alpha+\beta) - l\dot{\theta}\cos\beta,$$

and

$$0 = (\dot{x}\cos\theta + \dot{y}\sin\theta)\cos(\alpha+\beta) + (-\dot{x}\sin\theta + \dot{y}\cos\theta)\sin(\alpha+\beta) + l\dot{\theta}\sin\beta$$

The above two equations are the constrained condition of the differential deriven wheeled mobile robot.

With the constrained conditions of the mobile car and with the kinematic model, we are able to do the kinematical analysis of mobile car. Here we assume that the angle of wheels is $\theta_1(t)$ and $\theta_2(t)$ respectively, the radius of wheels is r, the driven ratio of gears is i where $i1$. Therefore the driving angular velocities are $i\theta_1(t)$ and $i\theta_2(t)$. We have,

$$V_1 = \dot{x}\cos\theta + \dot{y}\sin\theta \quad V_2 = -\dot{x}\sin\theta + \dot{y}\cos\theta \quad V_3 = l\dot{\theta}$$

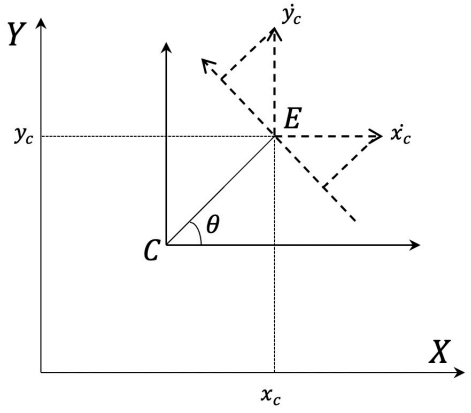


Figure 8: E velocity projection.

Therefore the constrained equation is changed into the following, analyse the left wheel first: Along the wheel plane we have $V_1 - V_3\cos\beta = ir\dot{\theta}_l$.

Simplify and we get $\dot{x}\cos\theta + \dot{y}\sin\theta - l\dot{\theta}\cos\beta = ir\dot{\theta}_l$.
In the plane perpendicular to the wheel we have $V_2 = V_3\sin\beta$,
Simplify and we get $l\dot{\theta}\sin\beta = -\dot{x}\sin\theta + \dot{y}\cos\theta$.
Similarly for the right wheel we have $\dot{x}\cos\theta + \dot{y}\sin\theta + l\dot{\theta}\cos\beta = ir\dot{\theta}_r$.
With the above equations, we have the kinematic equation of the differential driven mobile robot:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \frac{ir}{2} \begin{pmatrix} \frac{\cos(\theta-\beta)}{\cos\beta} & \frac{\cos(\theta+\beta)}{\cos\beta} \\ \frac{\sin(\theta-\beta)}{\cos\beta} & \frac{\sin(\theta+\beta)}{\cos\beta} \\ -\frac{1}{l\cos\beta} & \frac{1}{l\cos\beta} \end{pmatrix} \begin{pmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{pmatrix}$$

With the above equation, we have the following observation:

- If $\theta \neq 0$, then the motion constrain equation is not possible to be integrated into finite form, so the mobile robot is a **non-holonomic constrained** system.
- According to the characteristics of the non-holonomic constraint system, given any motion state of the mobile trolley, it is not always possible to obtain the angular velocities of left and right wheel respectively. But inversely, according to the left and right wheel angular velocities, we can confirm the state of motion of the car.

Next we discuss the form of non-holonomic constrained of the mobile manipulator.

Take C as the base point and ray CE as the baseline. Because the plane motion of a rigid body can be decomposed into the translation of the translational coordinate system with the base point as the origin and the fixed-axis motion of the rigid body relative to the translational coordinate system. So there is the following relationship:

$$v_E = v_r + v_C,$$

which is the linking equation between the velocities of C and E. Since at any moment the platform moves along the direction of CE, the sum of the velocity vectors on the transverse axis of symmetry passing through point C and point E is zero.

If we do the projection of the above equation on the symmetry axis, we get:

$$-\dot{x}_E\sin\theta + \dot{y}_E\cos\theta = L\dot{\theta}$$

Simplify and we get:

$$\dot{x}_E\sin\theta - \dot{y}_E\cos\theta + L\dot{\theta} = 0,$$

which can describe the kinematic equation of mobile manipulator under non-holonomic constraint.

If we take the joint variables as $q = [\theta_1, \theta_2, \theta 3, x, y, \theta]^T$, $A(q) = [0, 0, 0, \sin\theta, -\cos\theta, L]$. Here q is the generalized coordinate, then the constrained can be represented as:

$$A(q)\dot{q} = 0$$

Also, we can sum the equations of left and right wheels as:

$$2\left(\dot{x}_E\cos\theta + \dot{y}_E\sin\theta\right) = r\left(\dot{\theta}_r + \dot{\theta}_l\right)$$

Also with the constrained we can solve that

$$\begin{aligned}\dot{x}_E &= \left(\tfrac{r}{2}\cos\theta + \tfrac{L\cdot r}{d}\sin\theta\right)\dot{\theta}_l + \left(\tfrac{r}{2}\cos\theta - \tfrac{L\cdot r}{d}\sin\theta\right)\dot{\theta}_r \\ \dot{y}_E &= \left(\tfrac{r}{2}\sin\theta - \tfrac{L\cdot r}{d}\cos\theta\right)\dot{\theta}_l + \left(\tfrac{r}{2}\sin\theta + \tfrac{L\cdot r}{d}\cos\theta\right)\dot{\theta}_r\end{aligned},$$

which can also be represented in the matrix form:

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ -\frac{L\cdot r}{d} & \frac{L\cdot r}{d} \end{bmatrix} \begin{bmatrix} \dot{\theta}_l \\ \dot{\theta}_r \end{bmatrix}$$

5.2. Controller Design and Stability Analysis

For the constrained control part, we design a position control loop. The outer loop is responsible for maintaining position and we use a proportional controller based on the error between actual and demanded position to compute the desired speed of the motor. The error of system and sliding mode variable:

$$\begin{aligned} e(t) &= X_d(t) - X(t) \\ S(t) = \dot{X}_r(t) - \dot{X}(t) &= \dot{e}(t) + \Lambda e(t)\end{aligned}$$

The control law based on the above analysis is:

$$T_X = Ks + w + u$$

Fisrt we set the Lyapunic function is:

$$V = \frac{1}{2}s^T M_X(q)s$$

Do time differential and plug variables in:

$$\begin{aligned}\dot{V} &= \tfrac{1}{2}s^T\dot{M}_X(q)s + s^T M_X(q)\dot{s} \\ &= s^T\left(\tfrac{1}{2}\dot{M}_X(q) - C_X(q,\dot{q})\right)s + s^T\left(\Delta_X(q,\dot{q}) - Ks - u\right) \\ &= -s^T Ks - s^T u + s^T\Delta_X(q,\dot{q}) \\ &\le -s^T Ks - s^T\tfrac{s\rho^2(e,\dot{e})}{\|s\|\rho(e,\dot{e})+\varepsilon} + \|s\|\rho(e,\dot{e})\end{aligned}$$

Based on the property of Cauchy inequality $|x^T y| \le \|x\|\cdot\|y\|$, the above can be transformed into:

$$\begin{aligned}\dot{V} &\le -s^T Ks + \frac{\|s\|\rho^2(e,\dot{e})}{\|s\|\rho(e,\dot{e})+\varepsilon}\cdot\varepsilon \\ &\le -\lambda\|s\|^2 + \varepsilon\end{aligned}$$

From the stability theory of Lyapunov, we know that $\|\Delta(q,\dot{q})\| \le \rho(e,\dot{e})$ is correct.

Based on the above derivation, we set the Lyapunov function for our system as below:

$$V = \frac{1}{2}s^T M_X s + \frac{1}{2}\sum_{k=1}^{n}\tilde{\theta}_k^T\Gamma_k^{-1}\tilde{\theta}_k + \frac{1}{2}\sum_{k=1}^{n}\tilde{\alpha}_k^T Q_k^{-1}\tilde{\alpha}_k + \frac{1}{2}\sum_{k=1}^{n}\tilde{\beta}_k^T N_k^{-1}\tilde{\beta}_k$$

Assume $k_r > \|E\|$, then we can simplify and get the expression:

$$\begin{aligned}\dot{V} &\leq -s^T \left(Ks + \Delta_X - u\right) \\ &\leq 0\end{aligned}$$

which is similar to the proof above.

According to the Lyapunov function stability theory, we can say that the controller is reasonable and satisfies the requirement of design.

6. Comparison Result and Analysis

Now we developed two different control methods for autonomous robot to track any given trajectory. The first method is PD control and the second method is constrained tracking control. In this section, we design a experiment to compare the PID control and the constrained control performance for trajectory tracking task. We use MATLAB software to do the simulation for the autonomous robot.

In the experiment, we control the autonomous robot to track the trajectory below with the two different control method.

$$\begin{cases} x = 25t + 50 \\ y = 25t \qquad\qquad 0 \le t \le 20 \\ z = 0 \end{cases}$$

We set the sample interval as 0.001s and the initial joint variables as $[0, 0, 0, 0, 0, 0]$. So the initial configuration of the autonomous robot is as below:

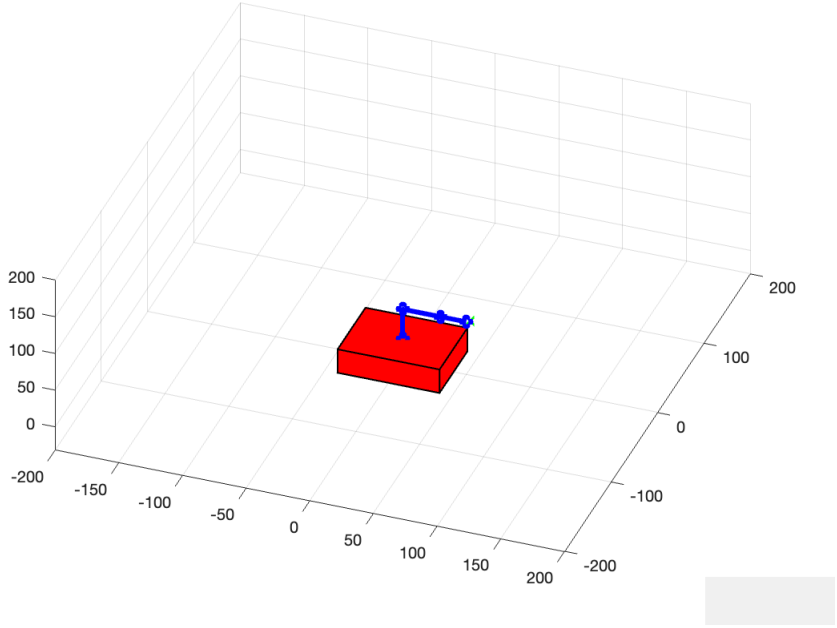
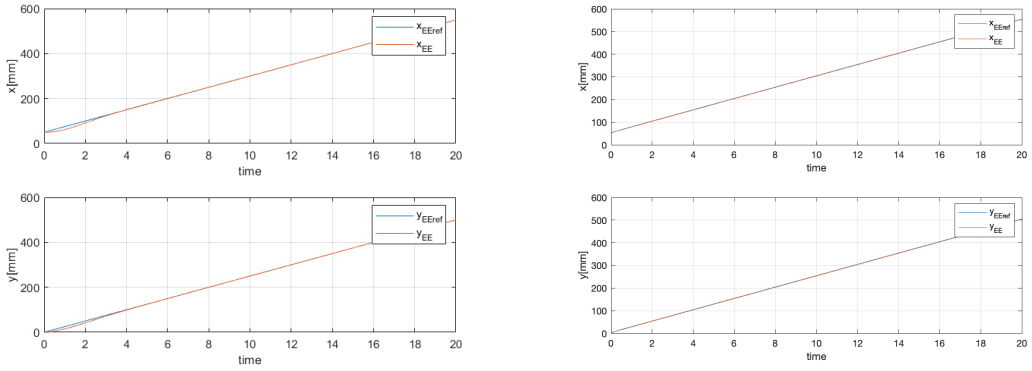


Figure 9: Initial configuration of the autonomous robot

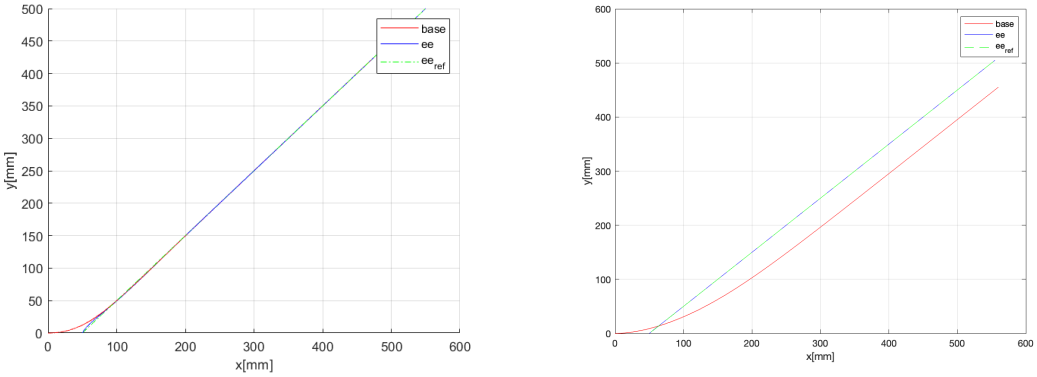
We will compare the trajectory and (angular) velocity of each joint. Firstly, we compare the true trajectory of the end-effector.



(a) PID control (b) Constrained control

Figure 10: True trajectory of the end-effecotr

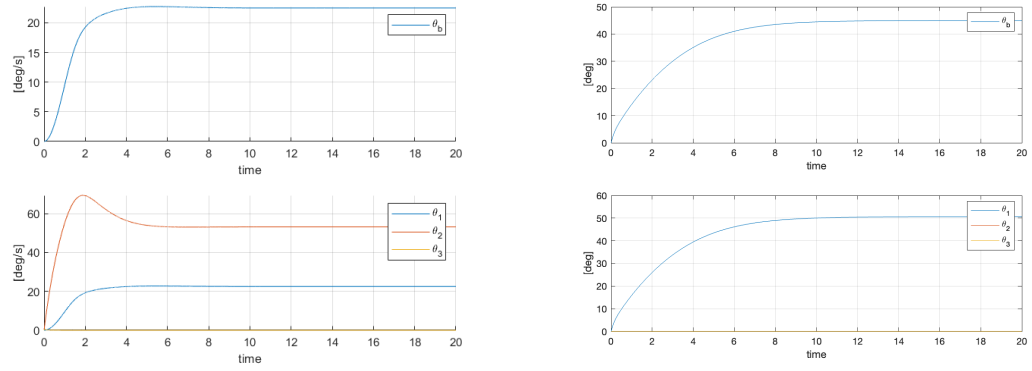
The projection of these two trajectories are as below:



(a) PID control (b) Constrained control

Figure 11: The projection of the trajectory

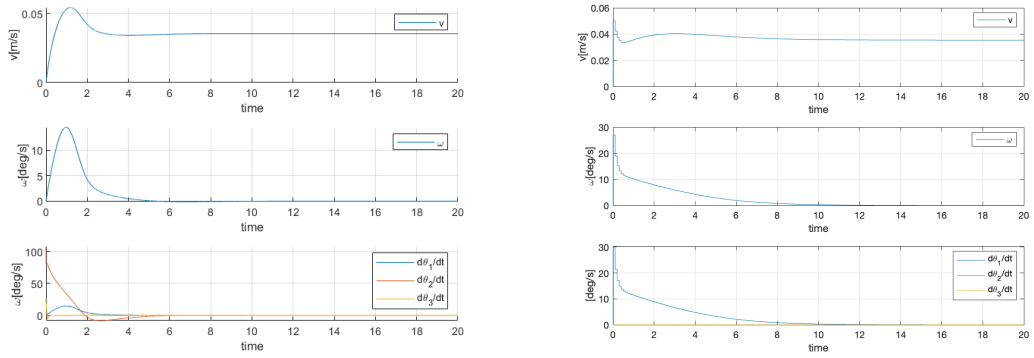
We further display the joint variables versus time.



(a) PID control (b) Constrained control

Figure 12: The joint variables versus time

Furthermore, we plot the (angular) velocities of the joints versus time.



(a) PID control (b) Constrained control

Figure 13: The (angular) velocities versus time

From the comparison above, we can see the different performance between the pure PID control and the constrained control for tracking the same one trajectory. Although the the steady-state error converges to both of the two control method, the constrained control shows advantage on the overshoot and the pure PID control has better performance on the rising time. The pure PID control conduct drastic change on the (angular) velocity at the beginning but converge to the steady state soon, while the constrained control lead to a smoother but slower process to the steady state. In all, the PID controller reflects larger overshoot but faster response time, while the constrained control has more stable response, therefore it gets more and more closer to the target trajectory but not strictly follow it, with an acceptable error.

In general, too large a proportional coefficient will cause the system to have a relatively large overshoot, and cause oscillations, which will deteriorate the stability. Here we discuss a little bit about the choice of gains. The main function of the differential link is to restrain the deviation from changing in any direction during the response process. The differential constant cannot be too large, otherwise the response process will be braked earlier and the adjustment time will be prolonged. What's more, the differential link will improve the anti-disturbance ability of the system and reduce the anti-noise ability of the system. Back to this problem, the movement of the base is relatively sensitive in the constrained control model. Therefore we should actually reduce the proportional gain of it. As for the pure PID control, the gains are selected according to experience.

7. Adaptive Neural Network Control of Simplified Manipulator

In this chapter, aiming at the wheeled mobile manipulator control system in an uncertain environment with unknown disturbances, a trajectory tracking control tracking method combining adaptive control and neural network control is proposed. This method is based on the dynamic model of the non-integrated wheeled mobile manipulator, and processes the internal parameters of the wheeled mobile manipulator system online through an adaptive algorithm. It appears in the control system through the neural network control algorithm and the sliding mode control algorithm. The error and chattering generated. Furthermore, the influence of unknown external interference and adaptive neural network approximation error on the system is eliminated, and the robustness and control system performance of the system are improved. The stability of the designed controller was proved by Lyapunov, and the simulation experiment verified that the designed controller can effectively realize the trajectory tracking control of the wheeled mobile manipulator.

7.1. Introduction of RBF NN-based Adaptive Control

RBF neural networks have shown much attention due to their good generalization ability and a simple network structure which avoids unnecessary and lengthy calculation. Since the principle of RBF NN is quite complicated and we can directly use it, we focus on the eooro dynamics and controller design part.

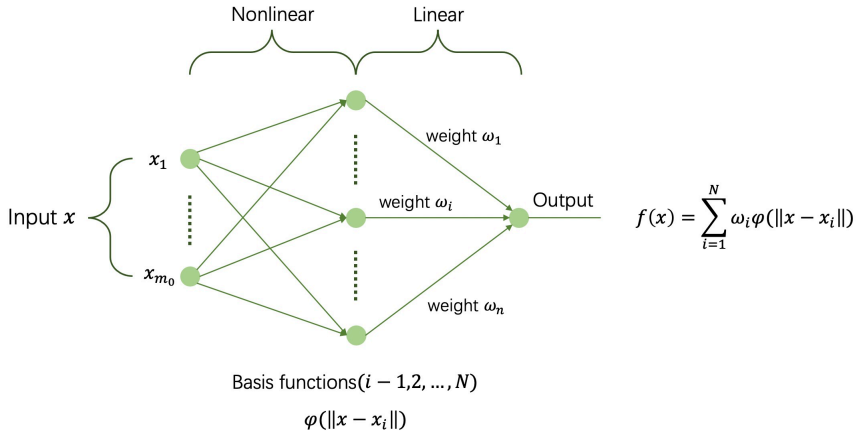


Figure 14: Radial basis function network structure.

7.1.1. Feedback Linearization

Feedback linearization is approximated by defining the following control input signal:

$$u = \widehat{h}^{-1}(y, v)$$

where v is referred to as a pseudocontrol. The function $\widehat{h}(y,v)$ represents the best available apprexi-mation of $h(y,u)$. Then, the system dynamics can be expressed as

$$y^r = v + \delta$$

where

$$\delta(\xi, v) = h\left(\xi_1, \widehat{h}^{-1}(\xi_1, v)\right) - \widehat{h}\left(\xi_1, \widehat{h}^{-1}(\xi_1, v)\right)$$

is the inversion error. r represents the r^{th} derivative of the input signal y. Then, the dynamics is reduced to

$$y^{(r)} = y^{*(r)} + L_d^c - a_c^s + \delta$$

7.1.2. Linear Compensator and Error Dynamics

Define the output tracking error as $e = y^* - y$. Then the dynamics is further written as:

$$e^{(r)} = -L_d^c + a_c^s - \delta.$$

Consequently. the error dynamics reduce to

$$e^{(r)} = -L_d^c$$

The following linear compensator is introduced to stabilize the dynamics

$$\begin{cases} \dot{\chi} = A_m\chi + b_m e \\ L_d^c = c_m\chi + d_m e. \quad \chi \in \mathfrak{R}^{r-1} \end{cases}$$

The dynamics referred to as tracking error dynamics is:

$$\begin{cases} \dot{E} = A_b E + b_b\left[a_c^s - \delta\right] \\ z = C_b E \end{cases}$$

where z is the vector of available measurements.

7.2. Adaptive NN Controller on Simplified Manipulator

With the RBF neural networks, it is easy for us to deal with uncertain parameters. However, time limited and the problem is too complicated, we reduce this system to a 2-DOF problem. The reason here is that for the mobile manipulator, actually the last link is not that useful if we go back to review the result mentioned in the last problem. We consider the dynamic of the 2-link manipulator with external disturbance can be expressed in the Lagrange as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau - T_d$$

We have obtained the symmetric inertial matrix $M(q)$, the vector of Coriolis and centripetal forces $C(q,\dot{q})$, and gravity vector $G(q)$ in the previous section. In this sub-problem, in order to simplify the problem, we set the frection $F(\dot{q})$ to be 0. T_d here represents the unknown disturbance input vector.

τ represents the joint toruqe input from motor.

The constraint equation can be expressed as $J^T(q)\dot{q} = 0$. Take the n-m -dimensional full-rank matrix $L(q)$ as a set of bases of the zero space of $J(q)$ like this:

$$L^T(q) \cdot J^T(q) = 0 \tag{32}$$

Introducing a transformation $\dot{q} = L(q)\dot{\beta}$ simplifies the system model to:

$$\bar{M}(q)\ddot{\beta} + \bar{C}(q,\dot{q})\dot{\beta} + \bar{G}(q) + \bar{\tau}_d = L^T\tau \tag{33}$$

where $\bar{M}(q) = L^T(q)M(q)L(q)$, $\bar{C}(q,\dot{q}) = L^T(q)V(q,\dot{q})$, $\bar{G}(q) = L^T(q)G(q)$, $\bar{\tau}_d = L^T(q)\tau_d$, $V(q,\dot{q}) = M(q)\dot{L}(q) + C(q,\dot{q})L(q)$.

To design a controller for a wheeled mobile manipulator with high control performance, it is assumed that the following properties are true.

- Exist α_1 and α_2, let $\bar{M}(q)$ is a positive definite inertia matrix and is uniformly bounded:

$$0 < \alpha_1 I \leq \bar{M}(q) \leq \alpha_2 I, \alpha_1 > 0, \alpha_2 > 0 \tag{34}$$

- $\dot{\bar{M}}(q) - 2\bar{C}(q,\dot{q})$ satisfies $x^T\left(\dot{\bar{M}} - 2\bar{C}\right)x = 0, \forall x \in R^n$.

- Through proper processing, the variables in the dynamic model of the wheeled mobile manipulator are linearized into the following equation:

$$M(q)\ddot{\rho} + C(q,\dot{q})\dot{\rho} + G(q) = \psi(q,\dot{q},\rho,\dot{\rho})\Theta \tag{35}$$

where $\psi(q,\dot{q},\rho,\dot{\rho}) \in R^{n\times r}$ is the regression matrix, which is the joint variable of the known wheeled mobile manipulator. $\Theta \in R^r$ is the physical description of the wheeled mobile manipulator, which is an unknown certain vector.

Similarly, as introduced in the last sub-section, here we define the tracking error and the sliding mode function as follows:

$$e(t) = q_d - q$$
$$s(t) = \dot{e} + \lambda e$$

where $\lambda = diag(\lambda_1, ..., \lambda_n)$ is a diagonal positive definite matrix.

Also, it is typical to define an error metric $s(t)$ to be a performance measured one. When the sliding surface $s(t) = 0$, the sliding mode is governed by the following differential equation according to the theory of sliding mode control: $\dot{e} = -\lambda e$. The behavior of the system on the sliding surface is determined by the structure of the matrix λ. If the error $s(t)$ is smaller, the performance is better. Therefore, the dynamic matrix can be rewritten as:

$$M\dot{s} = M(\ddot{e} + \lambda\dot{e}) = M(\ddot{q}_d + \lambda\dot{e}) - M\ddot{q}$$
$$M\dot{s} = f(x) - Cs - \tau + T_d$$

where $f(x)$ is defined as:

$$f(x) = M(q)(\ddot{q}_d + \lambda\dot{e}) + C(q,\dot{q})(\dot{q}_d + \lambda e) + G(q) + F(\dot{q}).$$

For the dynamics of the 2-link manipulator, the adaptive law is represented as follows:

$$\tau = -\tau_s + Ks + \hat{f}(x)$$

where K is a positive definite matrix. $\hat{f}(x)$ is the approximation of the adaptive function $f(x)$. τ_s represents the sliding mode controller robust term that is used to suppress the effects of uncertainties and approximation errors. The robust compensator τ_s is designed as:

$$\tau_s = -K_s \operatorname{sgn}(s)$$

where K_s is defined as

$$K_s = \varepsilon_0 + \tau_d.$$

The problem is now to be considered is to design a controller with a good control effect for the dynamic model Equation 33 of the autonomous robot. Because the system parameters are unknown, and all signals are uniformly bounded, for all t that is not less than zero and tends to positive infinity, the controller design has to ensure that the wheels and joints' trajectory must follow the desired trajectory. The error tracking curve of the joint should converge to 0 as quickly as possible. Due to the existence of unknown disturbances in the outside world, to improve the accuracy and reliability of the wheeled mobile manipulator control system, an adaptive neural network control algorithm is used to approximate the unknown disturbances that may appear. The following defines the expressions of the error variable, and the sliding mode variable are as follows:

$$\begin{gathered} e_m = \beta_d - \beta \\ s = \dot{e_m} + \Lambda e_m = \beta_r - \beta \\ \dot{s} = \ddot{\beta}_r - \ddot{\beta} \end{gathered} \tag{36}$$

In the above equations, $\dot{\beta}_r = \dot{\beta}_d + \Lambda e_m$, $dotq_r \in R^{n-m}$, $e_m, s \in R^{n-m}$, $\Lambda \in R^{n-m}$ are all positive-definite matric.

Bring Equation 36 to Equation 33, the error equation of the dynamic system of the mobile manipulator is as follows:

$$\begin{aligned} \bar{M}\dot{s} &= \bar{M}\ddot{\beta}_r - \bar{M}\ddot{\beta} \\ &= \bar{M}\ddot{\beta}_r + \bar{C}\dot{\beta}_r + \bar{G} + \bar{\tau}_d - \bar{C}s - L^T\tau \\ &= L^T\left(ML\ddot{\beta}_r + M\dot{L}\dot{\beta}_r + CL\dot{\beta}_r + G\right) + \bar{\tau}_d - \bar{C}s - L^T\tau \end{aligned} \tag{37}$$

According to the Property 3, we can get equation like this:

$$M(q)L\ddot{\beta}_r + M(q)\dot{L}\dot{\beta}_r + C(q,\dot{q})L\dot{\beta}_r + G(q) = \Psi\left(q, \dot{q}, L\dot{\beta}_r, \frac{dL\dot{\beta}_r}{dt}\Theta\right) \tag{38}$$

In the natural environment, the physical parameters $\Theta \in R^r$ of the wheeled mobile manipulator are usually unknown. The regression matrix Ψ is the known function matrix, which is the wheeled mobile manipulator's generalized coordinates and its derivatives. However, for the model parameters, it is unrealistic for uncertain wheeled mobile manipulators to solve its dynamics model and derive Ψ. At the same time, unknown external disturbances must be considered. These factors make the wheeled mobile manipulator's trajectory tracking control very difficult and challenging. This part introduces

an adaptive neural network control algorithm to deal with the above problems. The following is the structure of the RBF neural network:

$$Y = U_{NN}(\theta, X, B, C) = \theta^T \sigma(X) \tag{39}$$

Assuming that Lagrange equation always has an unknown constant ε_{hN} for $\forall x_1 \in D_h, \forall x_2 \in D_f$ can let the approximation error of the NN system satisfies $\|\varepsilon_h(x_1)\| \leq \varepsilon_{hN}, \quad \|\varepsilon_f(x_2)\| \leq \varepsilon_{fN}$.

In summary, the control law of the wheeled mobile manipulator is designed as follows:

$$\tau = \hat{\theta}_h^T \xi_h(x_1) + K_d L s + \hat{\varepsilon}_{hN} \operatorname{sgn}(Ls) + \hat{\theta}_f^T \xi_f(x_2) + \hat{\varepsilon}_{fN} \operatorname{sgn}(Ls) \tag{40}$$

where $\hat{\theta}_h, \hat{\theta}_f, \hat{\varepsilon}_{hN}, \hat{\varepsilon}_{fN}$ are the estimated value of $\theta_h, \theta_f, \varepsilon_{hN}, \varepsilon_{fN}$ respectively. Then we can define estimation error like this:

$$\begin{gathered} \tilde{\theta}_h = \theta_h - \hat{\theta}_h \cdot \\ \tilde{\theta}_f = \theta_f - \hat{\theta}_f \\ \tilde{\varepsilon}_{hN} = \varepsilon_{hN} - \hat{\varepsilon}_{hN} \\ \tilde{\varepsilon}_{fN} = \varepsilon_{fN} - \hat{\varepsilon}_{fN} \end{gathered} \tag{41}$$

$\hat{\theta}_h, \hat{\theta}_f, \hat{\varepsilon}_{hN}, \hat{\varepsilon}_{fN}$ is updated by the following adaptive update law:

$$\begin{gathered} \dot{\hat{\theta}}_h = \Gamma_h \xi_h(x_1) s^T L^T \\ \dot{\hat{\theta}}_f = \Gamma_f \xi_f(x_2) s^T L^T \\ \dot{\hat{\varepsilon}}_{hN} = \eta_h \|L(s)\| \\ \dot{\hat{\varepsilon}}_{fN} = \eta_f \|L(s)\| \end{gathered} \tag{42}$$

where Γ_h, Γ_f are positive definite diagonal matrix, η_h, η_f are positive constant.

7.2.1. Analysis of Stability

Lyapunov choose the following formula to find the derivative of Lyapunov with respect to time:

$$V = \frac{1}{2} s^T \bar{M} s + \frac{1}{2} \operatorname{tr}\left(\tilde{\theta}_h^T \Gamma_h^{-1} \tilde{\theta}_h\right) + \frac{1}{2} \operatorname{tr}\left(\tilde{\theta}_f^T \Gamma_f^{-1} \tilde{\theta}_f\right) + \frac{1}{2\eta_h} \tilde{\varepsilon}_{hN}^2 + \frac{1}{2\eta_f} \tilde{\varepsilon}_{fN}^2 \tag{43}$$

Put the formula shown below into the above formula.

$$M_X(q)\dot{s} = -C_X(q, \dot{q}) s + M_X(q) \ddot{X}_r + C_X(q, \dot{q}) \dot{X}_r + G_X(q) + \Delta X(q, \dot{q}) - T_X \tag{44}$$

We can get equation like this:

$$\begin{aligned}
\dot{V} =& s^T \bar{M}\dot{s} + \frac{1}{2}s^T\dot{\bar{M}}s + \operatorname{tr}\left(\tilde{\theta}_h^T\Gamma_h^{-1}\dot{\tilde{\theta}}_h\right) + \operatorname{tr}\left(\tilde{\theta}_f^T\Gamma_f^{-1}\dot{\tilde{\theta}}_f\right) - \frac{1}{\eta_h}\tilde{\varepsilon}_{hN}\dot{\tilde{\varepsilon}}_{hN} - \frac{1}{\eta_f}\tilde{\varepsilon}_{fN}\dot{\tilde{\varepsilon}}_{fN} \\
=& s^T\left(\bar{M}\ddot{q}_r + \bar{C}\dot{q}_r + \bar{G} + \bar{\tau}_d - \bar{C}s - L^T\tau\right) + \frac{1}{2}s^T\dot{\bar{M}}s + \operatorname{tr}\left(\tilde{\theta}_h^T\Gamma_h^{-1}\dot{\tilde{\theta}}_h\right) \\
&+ \operatorname{tr}\left(\tilde{\theta}_f^T\Gamma_f^{-1}\dot{\tilde{\theta}}_f\right) - \frac{1}{\eta_h}\tilde{\varepsilon}_{hN}\dot{\tilde{\varepsilon}}_{hN} - \frac{1}{\eta_f}\tilde{\varepsilon}_{fN}\dot{\tilde{\varepsilon}}_{fN} \\
=& s^T\left(\bar{M}\ddot{q}_r + \bar{C}\dot{q}_r + \bar{G} - L^T\tau\right) + s^T\bar{\tau}_d - \frac{1}{2}s^T(\dot{\bar{M}} - 2\bar{C})s \\
&+ \operatorname{tr}\left(\tilde{\theta}_h^T\Gamma_f^{-1}\dot{\tilde{\theta}}_h\right) + \operatorname{tr}\left(\tilde{\theta}_f^T\Gamma_f^{-1}\dot{\tilde{\theta}}_f\right) - \frac{1}{\eta_h}\tilde{\varepsilon}_{hN}\dot{\tilde{\varepsilon}}_{hN} - \frac{1}{\eta_f}\tilde{\varepsilon}_{fN}\dot{\tilde{\varepsilon}}_{fN} \\
=& s^T L^T(h(x) - \tau) + s^T\bar{\tau}_d + \operatorname{tr}\left(\tilde{\theta}_h^T\Gamma_f^{-1}\dot{\tilde{\theta}}_h\right) + \operatorname{tr}\left(\tilde{\theta}_f^T\Gamma_f^{-1}\dot{\tilde{\theta}}_f\right) \\
&- \frac{1}{\eta_h}\tilde{\varepsilon}_{hN}\dot{\tilde{\varepsilon}}_{hN} - \frac{1}{\eta_f}\tilde{\varepsilon}_{fN}\dot{\tilde{\varepsilon}}_{fN}
\end{aligned} \tag{45}$$

After bring the system control law, adaptive update law and property 3 into the above formula, we can finally get the equation like this:

$$\begin{aligned}
\dot{V} =& s^T\left[-L^T K_d L s + L^T\varepsilon_h(x_1) - L^T\hat{\varepsilon}_{hN}\operatorname{sgn}(Ls)\right] \\
&+ s^T\left[L^T\varepsilon_f(x_2) - L^T\hat{\varepsilon}_{fN}\operatorname{sgn}(Ls)\right] + \operatorname{tr}\tilde{\theta}_h^T\left(\Gamma_h^{-1}\dot{\tilde{\theta}}_h + \varepsilon_h(x_1)s^T L^T\right) \\
&+ \operatorname{tr}\tilde{\theta}_f^T\left(\Gamma_f^{-1}\dot{\tilde{\theta}}_f + \varepsilon_f(x_2)s^T L^T\right) - \tilde{\varepsilon}_{hN}\|Ls\| - \tilde{\varepsilon}_{fN}\|Ls\| \\
&\leq -s^T L^T K_d L s + s^T L^T\varepsilon_h(x_1) + s^T L^T\varepsilon_f(x_2) - \|Ls\|\varepsilon_{hN} - \|Ls\|\varepsilon_{fN} \\
&\leq -s^T L^T K_d L s \\
&\leq 0
\end{aligned} \tag{46}$$

So when $0 \leq V(t) \leq V(0), \forall t \geq 0$, $V(t) \in L_\infty$ is valid. According to the Babalat theorem, when $\lim_{t\to\infty}\dot{V}(t) = 0$, we can get $\lim_{t\to\infty} s = 0$, which means the error vector converges to $s(t) = 0$. In addition, when $t \to \infty$, $e_m \to 0, \dot{e_m} \to 0$; based on the defination of $s, e_m, \dot{e_m}$, we can also know that when $t \to \infty$, $q \to q_d, \dot{q} \to \dot{q}_d$. Therefore, the adaptive neural network control system can ensure the stability and convergence of the system.

The architecture of the adaptive RBF NN is as follows:

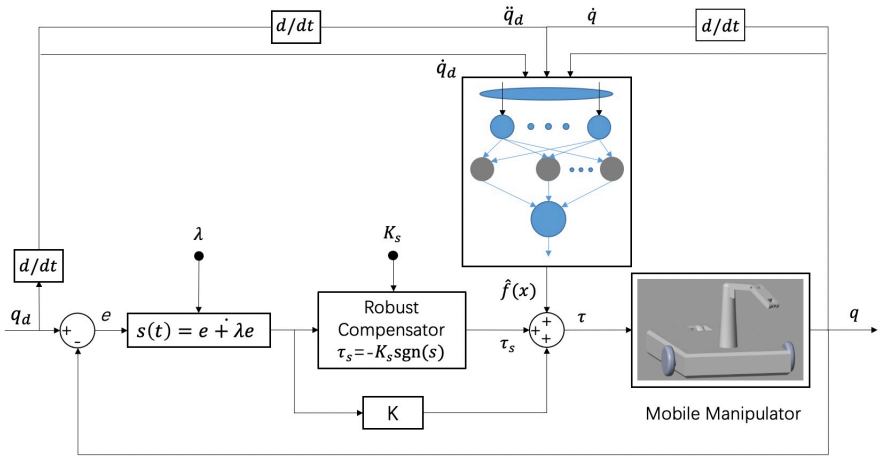


Figure 15: Structure of the whole control system.

7.3. Simulation

In this subsection we discuss the overall method and result of our model. In order to get the desired result, the steps are as follows:

- Train the network offline. The input is the desired trajectory $\theta_d, \dot{\theta}_d, \ddot{\theta}_d$, the output is τ_d .
- Put the obtained network into Simulink.
- The new input is the real $\theta_d, \dot{\theta}_d, \ddot{\theta}_d$ from the robots. We have the desired model which can be run in the Simulink.

The parameter values used in the adaptive control system are

$$\lambda = diag[5, 5], \quad K = diag[20, 20]$$

We choose the desired trajectory as: Q1=0.5*abs(a0*$S_0 + a21 * S2_1 + a22 * S2_2 + a31 * S3_1 + a32 * S3_2 + a33 * S3_3$); $Q2 = 0.4 * abs(a0 * S_0 + a21 * S2_1 + a22 * S2_2 + a31 * S3_1 + a32 * S3_2 + a33 * S3_3)$; $S_0 = H/2 * (1 - cos(pi * (t - t0_0)/(T0_1 - t0_0)))$; $S2_1 = H1/2 * (1 - cos(pi * (t - t1_0)/(T1_1 - t1_0)))$; $S2_2 = H1/2 * (1 - cos(pi * (t - T1_1 - T1_2)/(T1_1 - t1_0)))$; $S3_1 = H2/2 * (1 - cos(pi * (t - t2_0)/(T2_1 - t2_0)))$; $S3_2 = H2$; $S3_3 = H2/2 * (1 - cos(pi * (t - T2_1 - T2_2)/(T2_1 - t2_0)))$; $a0 = 0.6; a21 = 0.6; a22 = 0.55; a31 = 0.35; a32 = 0.75; a33 = 0.35$.

As for the RBF network, it is trained offline. Here, P_1, P_2 is the input parameters, T_1 and T_2 are the expected torques. Use genism to create the network: $\text{net}_1 = newrb(P1, T1, 1e-3, 1); a1 = sim(net_1, P1); net_2 = newrb(P1, T2, 1e-3, 1); a2 = sim(net_2, P1); gensim(net_1, -1) gensim(net_2, -1)$

Then the network 1 and network 2 are as follows:

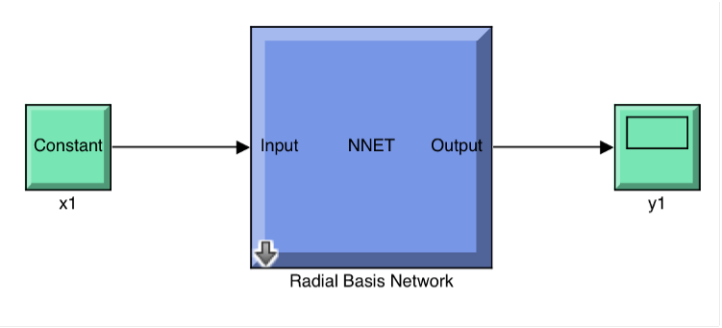


Figure 16: Network configuration.

7.4. Result

Run the simulation and we can get the following result:

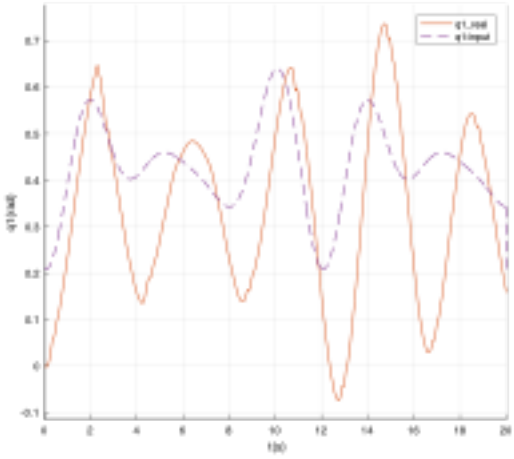


Figure 17: Desired vs. real q_1.

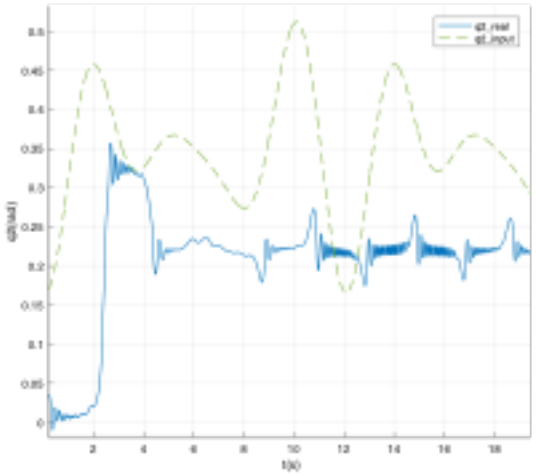


Figure 18: Desired vs. real q_2.

Actually as shown in the plotted graphs, the result of network control is not that good. There is obvious delay and sometimes oscillation. This is because we do not have enough time to try different parameters for the network as well as doing the adjustment. This can be a good attempt for us to try using the network on robotics problem though it is simplified.

8. Summary and Acknowledgement

8.1. Summary of Project

In this project, we searched a lot from all sources and did a lot of analysing and coding. In section one we did a literature review on autonomous robot design, virtual robot simulation, and constrained control. Then we derived the Lagrange-Euler equations for both robot arm and the whole autonomous robot, followed by a brief introduction of modeling and converting the model into Matlab.

In the next two sections both PID control and constrained control are introduced. Both ideas of designing the corresponding controllers are introduced, and stability analysis is done for the constrained control model. In the following section, the comparing experiment is done through simulation. We set the same trajectory for both controllers to control the trajectory of the end-effector. From the result we found that the constrained controller has less overshoot, smoother trajectory, but slower response time.

Next, the adaptive neural network control is introduced to control the simplified manipulator. For this subsection we used the RBF neural network in order to deal with the uncertain parameters in dynamic model. The whole model is run on Simulink platform. Overall, the model shows fine result on dealing with the uncertain parameters in dynamic environment. Time limited, some hyper-parameters cannot be adjusted well which causes the error on the result.

8.2. Acknowledgement

All the three of us want to sincerely thank Prof Ge first. Without his help, we could not finish the project. Actually it is the first time that all the three of us learn about control in robotics systematically. Prof Ge have got our interest in this topic and we will all work hard for it. Also, we need to send our thank to Kaiqi who helped us a lot during this period. In the future, we also hope to discover more about robotic control and dynamics part.

8.3. Decleration

Division of labor within the group:

- Sun Zhanhong: literature review on constrained control, constrained kinematics, constrained controller design, constrained model stability analysis, NN controller and stability analysis, result analysis
- Zhou Kunran: literature review, Lagrange-Euler equation detivation, CAD modeling to Matlab converting
- Cao Yuhong: CAD model design, Kinematics and PID control design, Comparing of PID and constrained control.
- All members help each other when needed.

References

[1] George A Bekey. *Autonomous robots: from biological inspiration to implementation and control*. MIT press, 2005.

[2] Kenzo Nonami y col. *Autonomous flying robots: unmanned aerial vehicles and micro aerial vehicles*. Springer Science & Business Media, 2010.

[3] Pratap Tokekar y col. «Tracking aquatic invaders: Autonomous robots for monitoring invasive fish». En: *IEEE Robotics & Automation Magazine* 20.3 (2013), págs. 33-41.

[4] Mohammad A Jaradat, Muath Bani-Salim y Fahed Awad. «A Highly-Maneuverable Demining Autonomous Robot: an Over-Actuated Design». En: *Journal of Intelligent & Robotic Systems* 90.1 (2018), págs. 65-80.

[5] John Von Neumann. «First Draft of a Report on the EDVAC». En: *IEEE Annals of the History of Computing* 15.4 (1993), págs. 27-75.

[6] Eugene Kagan, Nir Shvalb e Irad Ben-Gal. *Autonomous Mobile Robots and Multi-Robot Systems: Motion-Planning, Communication, and Swarming*. John Wiley & Sons, 2019.

[7] Song Yixu, Tan Dalong y col. «Velocity estimation of wheeled mobile robot with inner-sensors». En: (2003).

[8] Norbert Wiener. «Cybernetics». En: *Bulletin of the American Academy of Arts and Sciences* 3.7 (1950), págs. 2-4.

[9] Rodney A Brooks. «New approaches to robotics». En: *Science* 253.5025 (1991), págs. 1227-1232.

[10] Ronald C Arkin, Ronald C Arkin y col. *Behavior-based robotics*. MIT press, 1998.

[11] Yi Yong y Song Xueping. «The Present Situation and Developing Trend of Robot Simulation Research». En: *Mechanical Engineer* 7 (2009), págs. 63-65.

[12] Ping Lu. «Constrained tracking control of nonlinear systems». En: *Systems & control letters* 27.5 (1996), págs. 305-314.

[13] N. H. McClamroch y D. Wang. «Feedback Stabilization and Tracking of Constrained Robots». En: *1987 American Control Conference*. 1987, págs. 464-469. DOI: 10.23919/ACC.1987.4789364.

[14] Fuchun Sun y col. «Neural network plus fuzzy PD control of tip vibration for flexible-link manipulators». En: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. Vol. 3. IEEE. 2004, págs. 2942-2947.

[15] Heidar Ali Talebi, Rajni V Patel e Hikmet Asmer. «Neural network based dynamic modeling of flexible-link manipulators with application to the SSRMS». En: *Journal of Robotic systems* 17.7 (2000), págs. 385-401.

[16] Wenjie Dong. «On trajectory and force tracking control of constrained mobile manipulators with parameter uncertainty». En: *Automatica* 38.9 (2002), págs. 1475-1484.

[17] Dong Xu y col. «Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network-based sliding mode approach». En: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.3 (2009), págs. 788-799.

[18] KV Shihabudheen y col. «Stability control and trajectory tracking of two wheeled mobile manipulator». En: *2015 Annual IEEE India Conference (INDICON)*. IEEE. 2015, págs. 1-6.

[19] Kimitoshi Yamazaki y Masayuki Inaba. «Trajectory control of wheeled mobile robots based on virtual manipulators». En: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2009, págs. 2973-2978.