

Spider Lawn Mower Automation Project Proposal

Alfredo Chávez Plascencia

January 25, 2017

Abstract

This project proposal describes the automation of a four steering wheel lawn mower robot called spider lawn mower SLM robot. The SLM faces four issues called case studies. The first case study has to do with slippage conditions and follow cutting grass edge with no overlapping a ditch cutted grass. The second study case has to do with coordination control of multi-mobile robot systems. The third case study has to do with a body detection within the grass in front of the robot's path. The fourth case study has to do with repeating the same task over and over till the operator puts and end. To this end, a mathematical model that takes into account the kinematics and dynamics of the SLM is proposed. Moreover, state feedback linearization is suggested as a control strategy to tackle the nonlinearities of the system.

Contents

I Preliminaries	3
1 Introduction	4
2 System Description	6
2.1 Sensors	6
2.1.1 The Cameleon3	6
2.1.2 The Bumblebee2	6
2.1.3 The SP1 stereo vision system	7
2.1.4 The DUO MLX Stereo Vision System	8
2.1.5 The FX6 3D LIDAR sensor	8
2.1.6 GPS receiver	9
2.1.7 Omron D6T-44L-06 MEMS Thermal Sensor	9
2.2 System Hardware	10
2.2.1 Asus	10
2.2.2 NUC	11
2.3 System Software	11
II Modelling	12
3 Kinematic Model	13
3.1 Preliminaries	13
3.1.1 Geometry of a Four Steering Wheel Mobile Robot	13
3.1.2 Constraint Matrix	14
3.1.3 Center orientable wheel	14
3.1.4 Rolling Without Slipping	14
3.1.5 No lateral Movement	14
3.2 Kinematic Model	15
4 Dynamic Model	17
4.1 Lagrangian formalism	17
4.2 Kinetic Energy of the Wheel	17
4.3 Kinetic Energy of the Frame	18
4.4 Total Kinetic Energy of the SLM's Frame	19
4.5 Dynamic Equations and Removal of Lagrangian Multipliers	20
5 Control by Feedback Linearization	22
5.1 Input-Output Feedback Linearization	22
5.2 Control Simulation Results	26
6 Motor Control	29
6.1 DC Motor model	29
6.1.1 Electrical Part	29
6.1.2 Mechanical Part	29
6.1.3 Mathematical Model	30
6.1.4 Control Design	30
6.2 Motor Control Simulation	32

7	Digital Control	34
7.1	Nonlinear System discretization	34
7.2	DC Motor controller Digitization	35
7.3	Digital Motor Control Simulation	37
7.4	ROS-RVIZ System Simulation	38
7.5	Conclusion	39
III	Case Studies	42
8	Case Study One:	
	Move Control Correction and Optimization	43
8.1	Slippage Conditions	43
8.2	Following cutted grass edge with no overlapping	43
	8.2.1 Ground Detection	43
	8.2.2 Row Detection	44
	8.2.3 Row Control	44
9	Case Study Two:	
	Coordination Control of Multi-Mobile Robot Systems	45
9.1	Master slave	45
9.2	Virtual structure	45
9.3	Behavior-based	45
9.4	Motion planning	47
10	Case Stdudy Three:	
	Living Person Detection	48
11	Case Study four:	
	Repeat Known Mission	49
A	Motor Parameters	53
A.1	Electrical Resistance R_m	53
A.2	Motor Torque K_t and Electromotive force constants K_e	53
A.3	Friction Coeficient b_m	54
A.4	Inertia of the Rotor J_m	54
A.5	Electrical Inductance L_m	54

Part I

Preliminaries

Chapter 1

Introduction

Mobility and controllability of mover loan machines (MLM) is still an immature open research field [1]. A brief review of the state of the art of MLM is carried out in [1] where it states that [2] proposes a design of a mobile robot which lacks discussion of mowing operation. Moreover localization, static and dynamic obstacle avoidance are overlooked. And, the system is controlled manually lacking artificial intelligence. [3] suggests an optimal route planning of a mobile robot where it does not converge autonomously to the path. Moreover, obstacle avoidance algorithm would work for static obstacles not for dynamic ones. The fundamental idea of a mobile robot is discusses in [4] lacking simulation and implementation results. The work done in [1] suggests a motor schema architecture to control a MLM and also the system suffer from slippage.

In our point of view, the systems mentioned previously lacks a complete mathematical formulation of a MLM that can potentially allow the system to be controlled by some classical control strategies. And, they also lack slippage uncertainties than can be added to the mathematical formulation model of the system.

The objective of this proposal is to cope with some of the drawbacks presented in the previous works and also add new more functionality to the system. The system in turn is the spider from [5] and shown in Figure 1.1.



Figure 1.1: The spider lawn mower robot

Thus, the proposal is focused on solving four issues called case studies which are stated in the following:

- Move control correction and optimization. This task has to do with the correction of slippage conditions and following cutted grass edge with no overlapping.
- Coordination control of multi robot systems.
- Body detection.
- Repeat known task.

The first two case studies are presented and analyzed in the chapters 8 and 9 respectively. The last two case studies are not presented because they are subtasks of the first two case studies and they can be presented and analyzed when the first two case studies are solved.

In order to tests the algorithms in a real environment and before they are transferred to the SLM robot, the andromina v.1.2 from [6] and depicted in Figure 1.2 can be used. This is a platform that can be adapted to any open source platform, like Arduino, Raspberry Pi, PICAXE, ROS etc. Moreover, multiple motors, servos and sensors can be placed to this platform. Table 1.1 shows the andromina specifications.



Figure 1.2: Andromina OFF road 1.2v robot.

Model	Andromina OFF Road v.1.2
Dimensions	33.8 × 23.0 × 15.0 cm
Weight	1.5 Kg
Load	0.5 Kg
Voltage	3 → 12.0 V
Compatibility	Arduino, Raspberry Pi, PICAXE, ROS
Unit price	€ 230.00
four encoders	€ 16.00
four supports	€ 16.00
four servos with feedback	€ 50.00
12v battery	€ 70.00
Total	€ 382.00

Table 1.1: Andromina v.1.2 specifications.

Chapter 2

System Description

2.1 Sensors

2.1.1 The Cameleon3

The vision system is perhaps the highest potential sensor used in mobile robots. But it is probably also the most difficult sensor to master. A common camera configuration used in mobile robots is stereo-couple or stereo vision system. Figure 2.1b shows a stereo vision system that has build up using a two single cameras.

The Cameleon3 from Point-Grey [7] as shown in Figure 2.6 has a CCD (Charge Couple Device) sensor that consists of numerous light-sensitive semiconductor elements called photosensors.



(a)



(b)

Figure 2.1: (a) Shows the Cameleon3 CM3-U3-1352M-CS CCD camera. (b) Shows a stereo vision system based on two single cameras.

These photosensors can be seen as a tiny rectangular blocks or pixels (an acronym for picture element). When the incoming photons reach the semiconductor material, electrons are produced. The number of electrons are proportional to the light which reaches the light-sensitive part of the sensor. These electrons are stored in a capacitor, which is connected to a MOS transistor acting as a light switch. These electrons contain certain voltage, which is called the video signal. The electrons have to be transported via shift registers to a frame grabber, where all the pixels are digitised i.e. the pixels are converted to integer values and arranged in a $M \times N$ matrix and stored in a memory buffer. Table 2.6 shows some of the technical specifications of the Cameleon3 camera.

2.1.2 The Bumblebee2

Figure 2.2 shows the stereo vision system Bumblebee2 from Poyntgray [8]. It is a a typical commercial stereo vision system used in various robotic projects []. Table 2.2 shows some of the technical specifications of bumblebee2.

Table 2.1: cameleon3-CCD technical specifications.

Manufacturer	Point-Grey
Model	CM3-U3-1352M-CS
fps (frame per second)	30
resolution	1288×964
Pixel size	3.75 μm
Data interface	USB 3.0
Sensor type	CCD
Mass	54.9 grams
Unit price	€ 219.00



Figure 2.2: The bumblebee2 stereo vision system.

Manufacturer	Point-Grey
Model	Bumblebee2 0.3 MP
Sensor Name	Sony ICX424
fps (frame per second)	48
Megapixels	0.3 MP
resolution	648×488
Pixel size	7.4 μm
Data interface	FireWire IEEE-1394a
Sensor type	CCD
Chroma	Mono
Power Requirements	12 V
Power Consumption (Maxi)	2.5 W at 12 V
Dimensions	157 mm x 36 mm x 47.4 mm
Mass	342 grams
Unit price	€ 1,590.00

Table 2.2: Technical specifications.

One of the drawbacks in 3D stereo vision systems is that they operate in a passive mode having difficulties in providing reliable data for navigating and mapping. Moreover, the precision and maximum depth is limited by the baseline between the cameras, and the quality of the distance values decreases very fast as depth increases. The advantages of a stereo vision system is the fact that they can provide rgb colors, their size is small and they have low price [9].

Other problem with conventional stereo systems is the noisy extracted data and also the instability in non textured environments where only few features can be detected. This makes stereo cameras useless in dark or non-textured environments where it is very difficult and sometimes impossible to make a 3D reconstruction of the objects in the scene.

2.1.3 The SP1 stereo vision system

Figure 2.3 shows the SP1 stereo vision system from [10]. The SP1 stereo vision system allows you to perform stereo matching on real-time camera imagery, which facilitates depth perception and 3D-reconstruction. Unlike most other approaches to depth sensing, such as structured light or time-of-flight measurements, stereo vision is a purely passive technology which also works in bright daylight. You can connect your own USB industrial cameras to the SP1 system, and thus maintain control over all parameters of the stereo vision setup. Using FPGA-technology and hardware implementations of state-of-the-art algorithms, the SP1 can process the camera images in real-time. The processing result, which is a sub-pixel accurate disparity map (an inverse depth map), is sent out over the gigabit-ethernet port to an attached computer or embedded system. Using our open-source and cross-platform API, this disparity map can be converted into a 3D point cloud [10]. Table 2.3 shows the technical specification of the SP1.



Figure 2.3: The SP1 stereo vision system.

Manufacturer	Nerian Vision Technologies
Model	SP1
fps	30 @ 640×480 and 20 @ 800×592 pixels
Power	< 4 V
Voltage	5 V DC
Dimensions	$10.5 \times 7.6 \times 3.6$ cm
Mass	0.25Kg
Interface	USB 2.0
Unit price	€2,200.00

Table 2.3: Technical specifications.

2.1.4 The DUO MLX Stereo Vision System

The DUO MLX [11] is an ultra-compact imaging sensor with a standard USB interface. Intended for use in research, industrial and integration, the camera's high speed and small size make it ideal for existing and new use cases for vision based applications.

With a programmable illumination board and built-in IR filters it allows for precise control of lighting environment. Delivering configurable and precise stereo imaging for robotics, inspection, microscopy, human computer interaction and beyond.



Figure 2.4: The DUO-MLX stereo vision system.

Manufacturer	DUO3D
Model	Duo-MLX
frequency	100 Hz
Dimensions	$52 \times 25 \times 13$ mm
Angle M8 Lenses	170°
Interface	USB 3.0
Unit price	€

Table 2.4: Technical specifications.

2.1.5 The FX6 3D LIDAR sensor

The Micro-Electro-Mechanical Systems (MEMS) mirror based FX6 3D light detection and ranging (LIDAR) sensor by Nippon Signal [12] is depicted in Figure 2.5.

The FX6 sensor measures the distance and reflectance intensity using an infrared pulsed laser light with up to 16 fps. It measures the time-of-flight at a smallest unit of about 30 picoseconds (ps), which leads to a ranging precision of at most 1cm and a range of 16m. The laser beam is reflected by a mirror oscillating independently into two directions and thereby creating a full scan in a single iteration. The resolution of a full scan is 59×29 pixels covering a field of view of $50^\circ \times 60^\circ$ (HFOV/VFOV). Moreover, The FX6 has a low power consumption of 6W and a weight of 1kg. The main advantage of the LIDAR sensor, compared with stereo vision, is the reliability under changing lighting conditions, e.g. the FX6 is not influenced by sunlight, therefore it can be used during the night and also in foggy environments meaning that it can be used 24 hours a day [13, 9]. Other advantage of LIDAR sensors over 3D stereo vision systems is that it computes the ranging information on the sensor, so no time consuming calculations has to be done on the computer side. Table 2.5 shows some of the technical specifications of the LIDAR FX6 sensor. Unfortunately at the time this proposal has been written, the FX6 has been discontinued

and the new product is the 3D MEMS InfiniSoleil FX8 [14] which cost is €7,000.00.



Figure 2.5: A 3D MEMS lidar (FX6 from Nippon Signal)

Manufacturer	Nippon signal
Sensor Name	Laser / FX6
Mode of operation	active (laser)
Range	16 m
Precision	80 mm ($\pm 3\sigma$)
Field of View	50 × 60 (HFOV/VFOV)
Output	x, y, z, intensity
fps (frame per second)	16
resolution	29×59 pixels
Data interface	USB2
Sensor type	CCD
Power Consumption	7.25 (12V, ca. 0.5A)
Dimensions	95 mm x 64 mm x 142 mm
Mass	1 Kg
Unit price	€ 5,000.00

Table 2.5: Technical specifications.

2.1.6 GPS receiver

Figure 2.6 shows Reach which is a GPS receiver that realize on real time kinematics (RTK) technology [15]. The Reach-RTK (RRTK) system can provide high precision position data at approximately 2cm accuracy.



Figure 2.6: The RTK-GPS receiver

And, in order to achieve high centimeter level accuracy two GPS units are used with one of them being stationary (base) and another one moving (rover) High. The high precision accuracy is needed for mapping and localizing with respect to the map. RRTK- system also uses Glonass, Beidou and QZSS which increases the amount of visible satellites resulting in a faster and better solution. It also uses and IMU device that not only determine position, but orientation as well. By fusing IMU with RRTK processing software phase tracking algorithms can be improved to work better. The INDIEGOGO company offers a suitable RRTK KIT which cost is \$570.00, and includes the following: 2 × Reach module, 2 × Tallysman multi-GNSS antenna, 2 × USB cable, 2 × USB-OTG cable, 2 × DF13 6P-to-6P, 2 × DF13 6P-to-jumper. For further details about RRTK refer to [15].

2.1.7 Omron D6T-44L-06 MEMS Thermal Sensor

The Omron D6T-8L-06 MEMS thermal sensor (OTS) [16] is a super-sensitive infrared temperature sensor that is able to detect the presence of stationary humans by detecting body heat unlike typical pyroelectric human presence sensors that rely on motion detection. And, can therefore be used by a mobile system to automatically take an action. Figures 2.7a, 2.7b and 2.7c show the OTS, the sensor's FOV and the heat detected body temperature respectively.

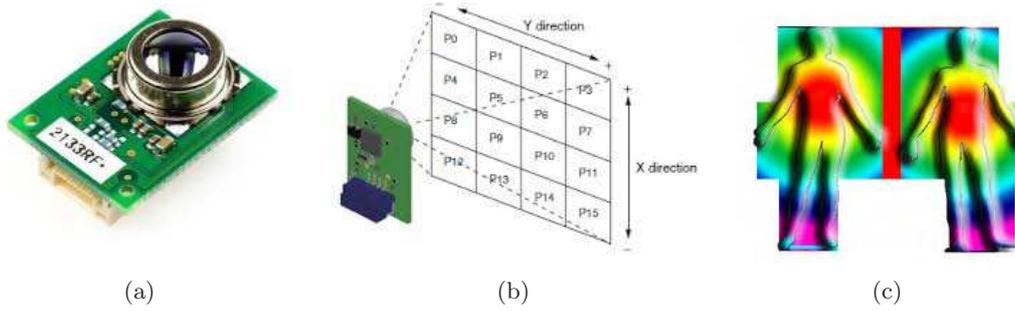


Figure 2.7: (a) The Omron D6T-8L-06 thermal sensor. (b) The thermal sensor's HFOV/VFOV. (c) The body heat detection.

Table 2.6: Omron D6T-8L-06 thermal Sensor.

Manufacturer	Mouser Electronics
Model	Omron D6T-8L-06
Field of View	$42.2^\circ \times 45.7^\circ$ (H/V)
Max Voltage	5.5 V
Min Voltage	4.5 V
Current	5 mA
Accuracy	$\pm 1.5\text{ C}, \pm 3\text{ C}$
Data interface	2-Wire, I2C, SMBus
Range	5-6 m
Unit price	€ 45.14

2.2 System Hardware

2.2.1 Asus

The Asus Z87I-Pro Mini-ITX board as depicted in figure 2.8 has been chosen mainly for the following reasons: it is economical, it is compatible with ROS and it is also compatible Linux system. The main characteristics of the board are presented in table 2.7. In fact there are other options in the market, but for a starting point this Mini ITX may be good enough and satisfy the requirements. The Dialog is open to suggest other boards.



Figure 2.8: Asus Z87I-Pro Mini-ITX.

Manufacturer	Asus
Model	Asus Z87I-Pro Mini-ITX
CPU Support	Intel i7 4770k Quad-Core
RAM	16GB DDR3-2400
Size	17.0 × 17.0 cm
Chipset	Intel Z87
USB	8 × USB 3.0 and 6 × USB 2.0
Sound	Realtek® ALC1150 8-Channel
Graphic	Dependant on installed CPU
Unit price	€ 185

Table 2.7: Technical specifications.

2.2.2 NUC



Figure 2.9: Intel NUC 6I5SYK

Manufacturer	Intel
Model	Intel NUC 6I5SYK
CPU Support	Intel Core i5
MAX RAM	32GB DDR4
Case Size	11.5 × 32 × 11.5 cm
Chipset	Intel
USB	4 × USB 3.0
Graphic	Intel HD Graphics
Unit price	€ 425

Table 2.8: Technical specifications.

2.3 System Software

Robot operating system (ROS) [17] is proposed as the software architecture to achieve the different case studies of the SLM.

The navigation stack (NS) that comes with the ROS installation is a set of configurable nodes that need to be configured properly to the shape and dynamics of the mobile robot in turn to be performed at a high level. Broadly speaking, the heart of the navigation stack is the move base node which provides a high level interface between odometer, robot base controller, sensors, sensor transforms, map server and monte carlo localization algorithm (AMCL) nodes to the local and global planners.

The global map is created by the gmapping package, which is based on simultaneous localization and mapping (SLAM). The gmapping data input stems from a range finder sensor. Then, during the functioning of the robot, the NS uses sensors to avoid obstacles on the path. And, also uses the sensors data to feed a costmap package to build a local map.

The localization and tracking position of the robot in the map is achieved by the AMCL node, which is a type of particle filter obtained by a proper substitution of the probabilistic motion and perceptual models into the algorithm of particle filter, [18].

To ensure a collision-free path planning, the NS uses the dynamic window approach planner (DWAP) and the Dijkstra's algorithm nodes. The DWAP restricts the admissible velocities to those that can be reached within a short time interval given the limited accelerations of the robot. Among the admissible velocities within the dynamic window the combination of translational and rotational velocity is chosen by maximizing an objective function, [19].

Thus, given a global path to follow the NS uses the costmap node that takes in the sensor data to build and inflate a local 2D occupancy grid map. This package also provides support to the DWAP that creates the velocity commands that drives the robot in the collision-free configuration space from a start to a final goal location.

The NS that comes with the ROS installation is well suited for indoor environments. However, when dealing with outdoor environments the rtabmap_ros package can be used [20] which is a ROS wrapper of RTAB-Map (Real-Time Appearance-Based Mapping), a RGB-D SLAM approach based on a global loop closure detector with real-time constraints [21]. This package can be used to generate a 3D point clouds of the environment and/or to create a 2D occupancy grid map for navigation. In other words, a stereo vision system can be used with the rtabmap_ros package to create a 3D map [20]. For navigation the move_base package can be used and the map_assembler node can generate a 2D occupancy grid from the point clouds [20].

If the system is equipped with a RRTK device, a high precision position data ($\sim 2\text{cm}$ accuracy) can be provided. This high precision can be used to reinforce the localization for mapping and navigation.

Part II
Modelling

frame with coordinate variables (x, y) . CM is the center of mass. P_c is the point of the center of mass with coordinates (x_c, y_c) . b is the distance between the center of each wheel and the geometry axis of symmetry. a is the length of the platform in the direction perpendicular to the driving wheel axis. l_i, α_i represents the fixed position of each wheel in the robot reference frame E . A_i is a single contact point of each wheel. r is the radius of each wheel. w_{fr} is the right front wheel. w_{fl} is the left front wheel. w_{br} is the right back wheel. w_{bl} is the left back wheel. θ is the angle of rotation.

3.1.2 Constraint Matrix

The constraint matrix of a mechanical system can be defined as follows.

Definition 3.1.1 (Constraint Matrix)[23]

Let q be n generalized coordinates subjected to m constraints such are in the form $C(q, \dot{q})$, with k holonomic constraint and $m - k$ nonholonomic constraints, all of which can be written in the form,

$$A(q)\dot{q} = 0 \quad (3.1)$$

where: $A(q)$ is an $(m \times n)$ full rank matrix.

In definition 3.1.1, $A(q)$ is the constraint matrix and some remarks can be stated.

Remark 3.1.1 • It is a holonomic constraint, if either a constraint equation is of the form $C(q)$ or it can be integrated.

• It is a nonholonomic constraint, if either a constraint equation is of the form $C(\dot{q})$, or it can not be integrated.

According to [22] the SLM has a centered orientable wheels subjected to two constraints.

3.1.3 Center orientable wheel

Also known as steering wheel. In this configuration the center of the wheel A is fixed in the body frame and its position with respect to the same frame is represented in polar coordinates $l \angle \alpha$. Also, the orientation of the wheel plane with respect to the frame E is given by the angle β which rotates about the vertical axle that passes through the center of the wheel. The position of this wheel with respect to the body frame is characterized by the following constants. l and α . And its motion by β and ψ as it is depicted in figure 3.2.

3.1.4 Rolling Without Slipping

This constrain assumes that the wheel has a single point contact with the ground without slipping. This means that the linear velocity of the wheel at the contact point must be zero, [22]. In other words, the velocity of the contact point V_A should be equal to the linear velocity of the wheel $r\dot{\psi}$. The coordinates of the contact point A with respect to the W reference frame are (a_x^W, a_y^W) and their linear velocities are $(\dot{a}_x^W, \dot{a}_y^W)$. This situation can be depicted in figure 3.2, and the constrain can be expressed mathematically as stated in equation 3.2.

$$[\cos(\beta_i) \quad \sin(\beta_i) \quad l_i \sin(\beta_i - \alpha_i)] R(\theta) \dot{\xi} - r \dot{\psi}_i = 0 \quad (3.2)$$

where; $i = 1, 2, 3, 4$, $R(\theta)$ is the 2D rotation matrix from frame $\{E\}$ to frame $\{W\}$ and ξ represents the robot pose at the point P_c . The following vectors are defined to ease the notation.

$$\xi = [x_c, y_c, \theta]^T, \quad \text{robot pose vector} \quad (3.3)$$

$$\beta_i = [\beta_1, \beta_2, \beta_3, \beta_4]^T, \quad \text{wheel steering angle vector} \quad (3.4)$$

$$\psi_i = [\psi_1, \psi_2, \psi_3, \psi_4]^T, \quad \text{wheel rotation angle vector} \quad (3.5)$$

3.1.5 No lateral Movement

This constrain assumes that the wheel's orthogonal components are zero, [22]. This situation is depicted in figure 3.3 and represented mathematically in equation 3.6.

$$[\sin(\beta_i) \quad -\cos(\beta_i) \quad -l \cos(\alpha_i - \beta_i)] R(\theta) \dot{\xi} = 0 \quad (3.6)$$

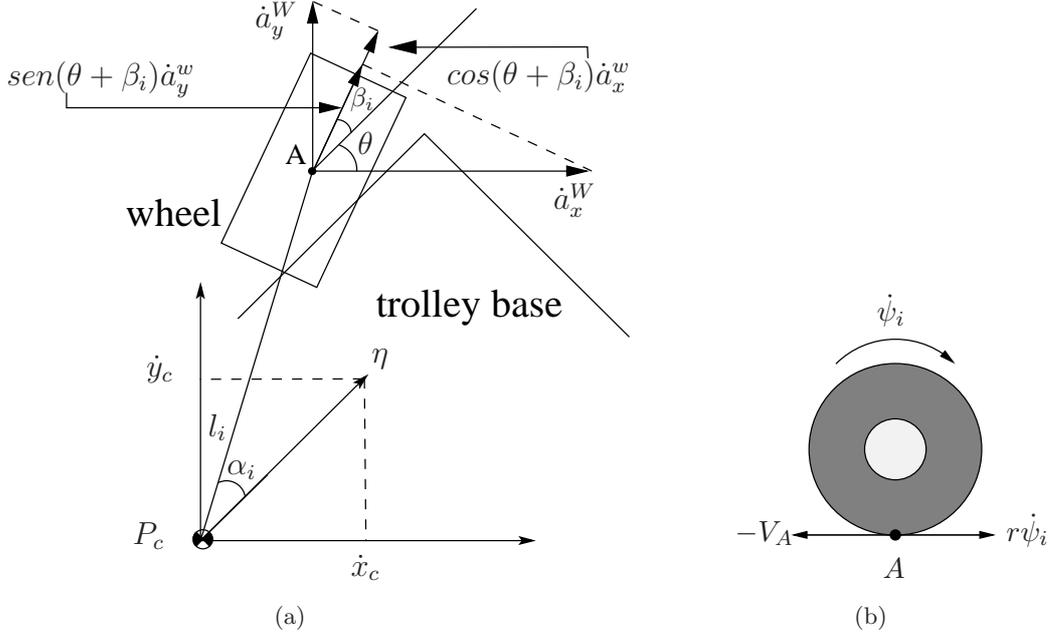


Figure 3.2: (a) Shows the rolling constraint without slipping. (b) Shows wheel with its linear and angular velocities

3.2 Kinematic Model

In the following a formal definition of kinematic is presented, furthermore a kinematic model of a SLM is shown.

Definition 3.2.1 (Kinematics)[24]

Kinematics is a branch of mechanics that has to do with the study and description of all possible motions of a rigid body

A kinematic model of a mechanical entity that has to do with the description of change of generalized coordinates q as function of velocities η is formally defined as follows

Definition 3.2.2 (Kinematic Model)[25]

From the mechanical system given by equation 3.1, Let $S_i = [s_1, \dots, s_{n-m}]^T$ be a set of smooth (continuously differentiable) and linearly independent vector fields in the null space of $A(q)$, ($null(A)$), such that, $A(q)S_i(q) = 0$, $i = 1, \dots, n - m$. Hence, $S_i(q) = \dot{q}$. Now it is possible to define $(n - m)$ velocities $\eta(t) = [\eta_1, \dots, \eta_{n-m}]^T$ such that for all t .

$$\dot{q} = S(q)\eta(t) \quad (3.7)$$

Equation 3.7 represents the kinematic model of a mechanical system, where $S(q)$ is a Jacobian mapping matrix from $\mathbb{R}^{(n-m)} \rightarrow \mathbb{R}^n$. In other words, it converts velocities from a mobile entity to velocities in a cartesian system.

In order to obtain a kinematic model of a SLM, firstly, a constraint matrix of the form of equation 3.1 must be obtained based on the constraints from equations 3.2 and 3.6. Equations 3.8 to 3.9 are the constraints imposed to the wheels of the system.

$$J_1(\beta)R(\theta)\dot{\xi} - J_2\dot{\psi} = 0 \quad (3.8)$$

$$C_1(\beta)R(\theta)\dot{\xi} = 0 \quad (3.9)$$

And they can be arranged in matrix form as shown in 3.10.

$$\begin{bmatrix} J_1(\beta)R(\theta) & 0 & -J_2 \\ C_1(\beta)R(\theta) & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\xi} \\ \dot{\beta} \\ \dot{\psi} \end{bmatrix} = A(q)\dot{q} = 0 \quad (3.10)$$

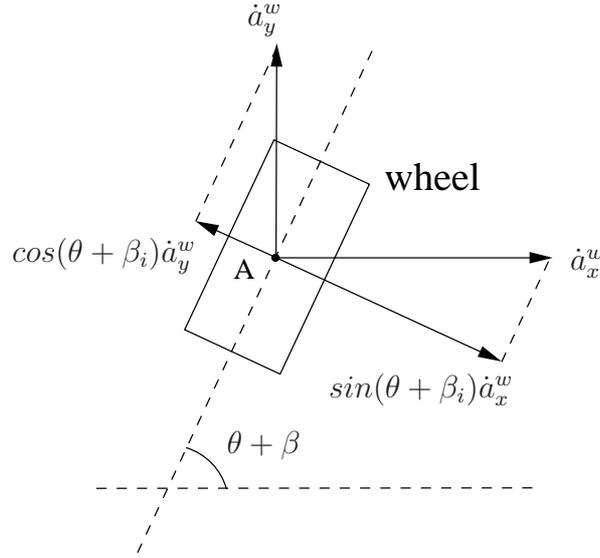


Figure 3.3: Shows the lateral constraint

Secondly, taking the no lateral movement constraint 3.9, it can be seen that $R(\theta)\dot{\xi}$ lies in the null space of $\mathcal{N}(C_1(\beta))$. Then, defining $\mathcal{N}(C_1(\beta)) = \Sigma(\beta)$ yields the following relation $\mathcal{N}(C_1(\beta))\eta(t) = \Sigma(t)\eta(t) = R(\theta)\dot{\xi}$. Afterwards, isolating $\dot{\xi}$ from the previous equation and then substituting it into equation 3.8 and defining the steering velocity vector $\dot{\beta} = \zeta = [\zeta_1, \zeta_2, \zeta_3, \zeta_4]$, the following kinematic model 3.11 which is of the form $\dot{q} = S(q)u(t)$ is obtained.

$$\begin{bmatrix} \dot{\xi} \\ \dot{\beta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} R^T(\theta)\Sigma(\theta) & 0 \\ 0 & I \\ J_2^{-1}J_1(\beta)\Sigma(\beta) & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \quad (3.11)$$

According to [22], in order to guarantee maneuverability (δ_M) of the SLM the degree of mobility δ_m and the degree of steerability δ_s equations 3.12 and 3.13 must satisfy the following conditions: $1 \leq \delta_m \leq 3$ and $0 \leq \delta_s \leq 2$.

$$\delta_m = \dim\mathcal{N}[C_1(\beta)] = 3 - \text{rank}[C_1(\beta)] \quad (3.12)$$

$$\delta_s = \text{rank}[C_1(\beta)] \quad (3.13)$$

Chapter 4

Dynamic Model

4.1 Lagrangian formalism

The relation between the torques τ derived by the embarked motors and the change in velocities \dot{u} in the SLM can be described by the dynamic model. The equations of motion that relates τ and \dot{u} can be derived by means of the Lagrangian formalisms. The Lagrangian formalism to holonomic and nonholonomic systems can be found in different sources in the literature [26, 24, 27] and stated in equation 4.1.

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right) - \frac{\partial L(q, \dot{q})}{\partial q} &= M_I(q)\ddot{q} + V(q, \dot{q}) \\ &= A^T(q)\lambda + B(q)\tau \end{aligned} \quad (4.1)$$

Where:

$L(q, \dot{q}) = T(q, \dot{q}) - W(q)$ is the Lagrangian as a function of the kinetic energy $T(q, \dot{q})$ minus the potential energy $W(q)$ of the system.

$M_I \in \mathbb{R}^{n \times n}$ is the inertia matrix of the system.

$V(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the centripetal and coriolis matrix.

$A(q)$ is the Jacobian transpose matrix of the constraint matrix.

$B(q) \in \mathbb{R}^{n \times (n-m)}$ is an input transformation matrix.

$\tau \in \mathbb{R}^{(n-m)}$ is the input torque vector.

$\lambda \in \mathbb{R}^m$ is the vector of constraint forces or the undetermined Lagrangian multipliers.

n is the number of generalised coordinates.

q are the generalised coordinates.

m the number of constraints.

Since most of the times the potential energy of the system remains constant it can be neglected from the Lagrangian multiplier. This fact lets L as a function of $T(q, \dot{q})$, i.e. $L(q, \dot{q}) = T(q, \dot{q})$.

In order to solve 4.1, the kinetic energy $T(q, \dot{q})$ of the system must be solved. First, a kinetic energy of the wheel with respect to CM is found. Then the kinetic energy of the main body of the wheelchair is solved and added to the kinetic energy of the wheels.

4.2 Kinetic Energy of the Wheel

Figure 4.1 shows the schematic representation of a single steering wheel attached to the trolley. It is assumed in this representation that the mass of the wheel is distributed evenly along the ring with radius r . Moreover, the point of contact of the wheel with the ground is supposed to be a single point. Hence, in order to find the kinetic energy of the wheel, a velocity of the point P_w with respect to a center of mass CM and which is relative to a fixed frame W is found, and then multiplied by the angular density ρ_α . The former can be established formally in the following theorem.

Theorem 4.2.1 (SLM's kinetic energy)

Let $P_w(x_{P_w}, y_{P_w}, z_{P_w})$ be a point on the surface of a wheel, $d_m[Kg]$ an infinitesimal mass of the point P_w , and $d_\psi[rad]$ and infinitesimal angle and $\rho_\alpha[Kg/rad]$ be the angular density, such that:

$$T_w = \frac{1}{2} \int_0^{2\pi} (\dot{x}_{P_w}^2 + \dot{y}_{P_w}^2 + \dot{z}_{P_w}^2) \rho_\alpha d\psi \quad (4.2)$$

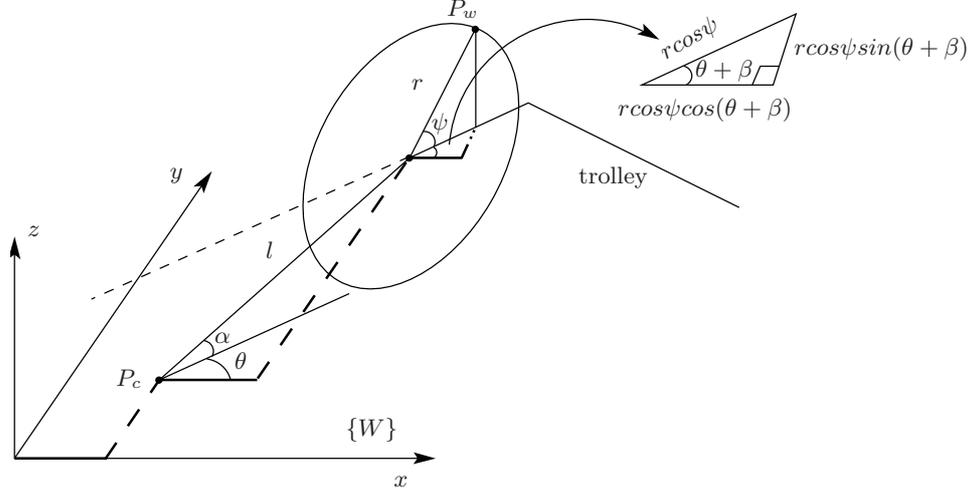


Figure 4.1: A wheel attached to the trolley. The mass m_w is assumed to be distributed evenly with radius r when computing the kinetic energy of the wheel with respect to the center of mass.

The position of the point $P_w = [x_{P_w}, y_{P_w}, z_{P_w}]^T$ with respect of the center of mass CM can be arranged in matrix notation.

$$P_w = \begin{bmatrix} x_c + l \cos(\alpha + \theta) + r \cos(\psi) \cos(\theta + \beta) \\ y_c + l \sin(\alpha + \theta) + r \cos(\psi) \sin(\theta + \beta) \\ r \sin(\psi) \end{bmatrix} \quad (4.3)$$

Taking the square derivative of \dot{P}_w^2 and substituting the result in 4.2 and then computing the integration, the kinetic energy of a wheel with respect to the center of mass is obtained.

$$T_w = \frac{1}{2} m_w \dot{x}_c^2 + \frac{1}{2} m_w \dot{y}_c^2 + m_w l \dot{\theta} (\dot{y}_c \cos(\alpha + \theta) - \dot{x}_c \sin(\alpha + \theta)) + \frac{1}{2} m_w l^2 \dot{\theta}^2 + \frac{1}{2} I_w \dot{\psi}^2 + \frac{1}{4} I_w \dot{\beta}^2 + \frac{1}{2} I_w \dot{\theta} \dot{\beta} + \frac{1}{4} I_w \dot{\theta}^2 \quad (4.4)$$

Where, I_w is the moment of inertia of the wheel and is defined as $I_w = m_w r^2$. m_w is the mass of the wheel and is defined as $m_w = 2\pi \rho_\alpha$.

4.3 Kinetic Energy of the Frame

By combining rotational and translational energies, a kinetic energy of the frame can be obtained. Figure 4.2 shows the SLM's frame rotational and translational velocities. And, assuming that the center of mass does not coincide with the center of the robot, then the frame's kinetic energy can be stated formally as in the following definition.

Definition 4.3.1 (Frame's kinetic energy)[24]

The kinetic energy of a rotating and moving frame is associated with the motion of its center of mass plus the rotational energy about its center of mass.

$$T_F = \frac{1}{2} M_T (\dot{x}_c^2 + \dot{y}_c^2) + \frac{1}{2} I_T \dot{\theta}^2 \quad (4.5)$$

$$\dot{x}_c = \dot{x} - d \dot{\theta} \sin(\theta + \rho) \quad (4.6)$$

$$\dot{y}_c = \dot{y} + d \dot{\theta} \cos(\theta + \rho) \quad (4.7)$$

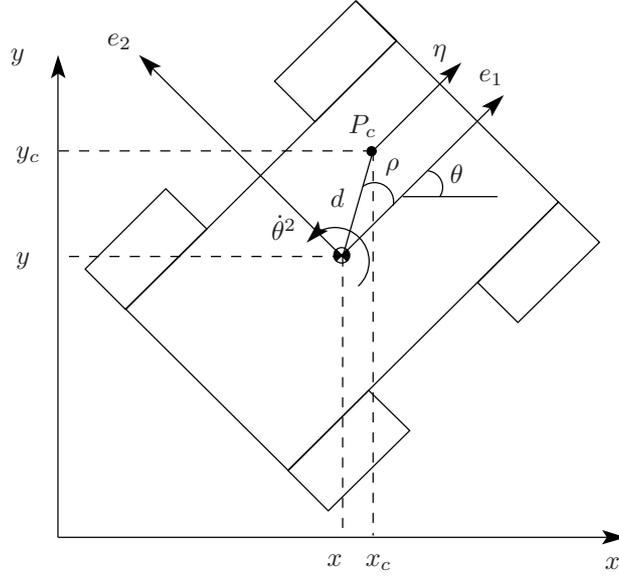


Figure 4.2: The SLM that rotates and translates about its center of mass.

Where:

M_T is the mass of the trolley or frame in [Kg].

$(\dot{x}_c^2 + \dot{y}_c^2)$ is the linear velocity of the trolley in [m/s] about its CM.

I_T is the moment of inertia of the frame in [Kg · m²] about its CM

$\dot{\theta}$ is the angular velocity in [Rad/s] of the frame about its CM

Then, inserting 4.6 and 4.7 into 4.5, the equation 4.8 is obtained which represents the kinetic energy of a rotational and moving SLM frame with respect to its center of mass CM.

$$T_F = \frac{1}{2}M_T\dot{x}_c^2 + \frac{1}{2}M_T\dot{y}_c^2 + \frac{1}{2}M_Td^2\dot{\theta}^2 + \frac{1}{2}I_T\dot{\theta}^2 - M_Td\dot{x}_c\dot{\theta}\sin(\theta + \rho) + M_Td\dot{y}_c\dot{\theta}\cos(\theta + \rho) \quad (4.8)$$

4.4 Total Kinetic Energy of the SLM's Frame

The total kinetic energy T of the SLM can be obtained by adding the kinetic energy of the frame T_F equation 4.5 plus the kinetic energy of the wheels T_w equation 4.4.

$$T = T_F + \sum_{i=1}^4 T_{w_i} \quad (4.9)$$

Substitution of equations 4.4 and 4.5 into equation 4.9 and then expanding it and arranging it into a matrix form gives equation 4.10 which is the total kinetic energy of the SLM with respect to its CM.

$$T = \frac{1}{2}\dot{\xi}^T R(\theta)^T M R(\theta) \dot{\xi} + \frac{1}{4}\dot{\beta}^T I_{WM} \dot{\beta} + \frac{1}{2}\dot{\psi}^T I_{WM} \dot{\psi} + \frac{1}{2}\dot{\theta} I_{WV} \dot{\theta} \quad (4.10)$$

Where :

$$\begin{aligned} I_{WM} &= \begin{bmatrix} I_w & 0 & 0 & 0 \\ 0 & I_w & 0 & 0 \\ 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & I_w \end{bmatrix} \\ I_{WV} &= [I_w \quad I_w \quad I_w \quad I_w] \\ M &= \begin{bmatrix} M_1 & 0 & M_2 \\ 0 & M_1 & M_3 \\ 0 & 0 & M_4 \end{bmatrix} \\ M_1 &= M_T + 4m_w \\ M_2 &= -2 \left[dM_T \sin(\rho) + m_w \sum_{i=1}^4 l_i \sin(\alpha_i) \right] \\ M_3 &= 2 \left[dM_T \cos(\rho) + m_w \sum_{i=1}^4 l_i \cos(\alpha_i) \right] \\ M_4 &= d^2 M_T + I_T + m_w \sum_{i=1}^4 l_i^2 + 2I_w \end{aligned} \quad (4.11)$$

4.5 Dynamic Equations and Removal of Lagrangian Multipliers

A procedure stated in [22] is followed in this proposal with emphasis in steering wheels. Thus, in order to derive the dynamic equations of motion of the SLM, the matrices $A^T(q)$ and $B(q) = [0_{3 \times 8}; \mathbb{I}_{8 \times 8}]$ are replaced in the right expression of equation 4.1. The Lagrange equations of motion of the SLM with Lagrangian multipliers λ_1 and λ_2 are given by equations 4.12 to 4.14.

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} = [T]_{\xi} = R^T(\theta) J_1^T(\beta) \lambda_1 + R^T(\theta) C_1^T(\beta) \lambda_2 \quad (4.12)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}} \right) - \frac{\partial T}{\partial \beta} = [T]_{\beta} = \tau_{\beta} \quad (4.13)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\psi}} \right) - \frac{\partial T}{\partial \psi} = [T]_{\psi} = -J_2^T \lambda_1 + \tau_{\psi} \quad (4.14)$$

The next step is to eliminate the Lagrange coefficients from equations 4.12 and 4.14. To do so, the equations 4.12 and 4.14 are premultiplied from the left with $\Sigma(\beta)R(\theta)$ and $\Sigma^T(\beta)J_1^T(\beta)J_2^{-T}$ respectively, and then summed up and taking into account that $\Sigma^T(\beta)C_1^T(\beta) = 0_{1 \times 4}$. This leads to two equations, from which the Lagrange coefficients have been banished. Furthermore, the total kinetic energy equation 4.10 is inserted in the left side of equations 4.12, 4.13 and 4.14 respectively. Once $[T]_{\xi}$, $[T]_{\beta}$ and $[T]_{\psi}$ are solved, they are inserted in the first two equations with no Lagrangian coefficients. This step leads up to two new equations that contain the velocity $\dot{\xi}$, $\dot{\psi}$, $\dot{\beta}$ and acceleration $\ddot{\xi}$, $\ddot{\psi}$, $\ddot{\beta}$ terms. Then, with the aid of the kinematic equations and their derivatives and substituting them back and arranged them in a matrix format yields the equation 4.15 which is the dynamic model of the system.

$$H(\beta)\dot{u} + f(\beta, u) = F(\beta)\tau \quad (4.15)$$

Where:

$$\begin{aligned}
\tau &= \begin{bmatrix} \tau_\psi \\ \tau_\beta \end{bmatrix} \\
F(\beta) &= \begin{bmatrix} \Sigma^T(\beta)E^T(\beta) & 0_{4 \times 4} \\ 0_{4 \times 4} & I_{4 \times 4} \end{bmatrix} \\
P &= \begin{bmatrix} M_1 & 0 & \frac{1}{2}(M_2 \cos(\theta) - M_3 \sin(\theta)) \\ 0 & M_1 & \frac{1}{2}(M_3 \cos(\theta) + M_2 \sin(\theta)) \\ \frac{1}{2}(M_2 \cos(\theta) - M_3 \sin(\theta)) & \frac{1}{2}(M_3 \cos(\theta) + M_2 \sin(\theta)) & M_4 \end{bmatrix} \\
E(\beta) &= J_2^{-1} J_1(\beta) \\
f(\beta, u) &= \begin{bmatrix} f_1(\beta, u) \\ f_2(\beta, u) \end{bmatrix} \\
f_1(\beta, u) &= \Sigma^T(\beta) \left[R(\theta) \dot{P} R^T(\theta) + R(\theta) P \dot{R}^T(\theta) - \frac{1}{2} R(\theta) K_1 \eta^T \Sigma^T(\beta) N \right. \\
&\quad \left. + E^T(\beta) I_{WM} \dot{E}(\beta) \right] \Sigma(\beta) \eta \\
&\quad + \Sigma^T(\beta) \left[R(\theta) P R^T(\theta) + E^T(\beta) I_{WM} E(\beta) \right] \dot{\Sigma}(\beta) \eta \\
f_2(\beta, u) &= \frac{1}{2} \ddot{\theta} I_w K_2 \\
H(\beta) &= \begin{bmatrix} H_1(\beta) & H_2(\beta) \\ 0_{1 \times 4} & \frac{1}{2} I_{WM} \end{bmatrix} \\
H_1(\beta) &= \Sigma^T(\beta) \left[R(\theta) P R^T(\theta) + E^T(\beta) I_{WM} E(\beta) \right] \Sigma(\beta) \\
H_2(\beta) &= \frac{1}{2} \Sigma^T(\beta) R(\theta) K_1 I_{wV}
\end{aligned}$$

The kinematics and dynamics of the SLM can be written in state space representation as stated in equations 4.16 and 4.17.

$$H(\beta) \dot{u} = -f(\beta, u) + F(\beta) \tau \quad (4.16)$$

$$\dot{q} = S(q) u \quad (4.17)$$

Chapter 5

Control by Feedback Linearization

This chapter deals with the control strategy of the SLM which is a system that is highly governed by nonlinearities. For this reason, a nonlinear control strategy has been chosen to tackle the implementation of the controller. More precisely, a state feedback linearization is of interest to handle the nonlinearities of the system. However, it has been proved that if one or more constraints in a nonlinear system are nonholonomic, the system is not fully input satiate linearizable. It may be input-output linearizable if a proper set of output equation are chosen, [25, 23, 28].

First at all, a mathematical formulation about input-output feedback linearization is introduced. Secondly, a study case of the SML is formulated.

5.1 Input-Output Feedback Linearization

In input-output feedback linearization, one of the keys is to find out if there exists a state transformation $z = T(x)$ and $x = T^{-1}(z)$ such that is a diffeomorphism, bringing the nonlinear system into a normal form. This form decomposes the nonlinear system into external and internal parts respectively, making the system partially linearizable. The external variables have a property that can be seen by the output, whereas the internal variables are hidden from the output. Moreover, there is a control law that will bring the external part of the normal system into a lineal controllable canonical form. The question now is whether or not the internal states will be bounded and stable. The problem of instability can usually be analyzed by the use of the zero dynamics of the system, [29, 30]. The relative degree of the system is a key factor in the application of state feedback linearization methods, which is in turn a chain of integrators that explicitly depends of the input. It may tell the number of outputs equations that must be chosen for a specific system.

In the following, a nonlinear system, a state transformation vector, a nonlinear form system, and a relative degree are presented as compact definition forms.

Definition 5.1.1 (Nonlinear System)[29]

Let a differential dynamic equation be represented as a single-input-single-output SISO system.

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \\ y &= h(x)\end{aligned}\tag{5.1}$$

Where, $f(x)$, $g(x)$ and $h(x)$ are assumed to be sufficiently smooth on the domain $D \subset \mathbb{R}^n$. $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}$ is the control input and $y \in \mathbb{R}$ is the output. The mappings $f : D \rightarrow \mathbb{R}^n$ and $g : D \rightarrow \mathbb{R}^{n \times p}$ are vector fields on D . And, assuming the system 5.1 has a relative degree ρ_r .

Definition 5.1.2 (State Transformation Vector)[29]

Let ρ_r be the relative degree of the system, n the total number of state transformation variables, ς represents the internal dynamics of the system and ξ_e represents the external variables, such that

$$z = T(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{n-\rho_r}(x) \\ \hline h(x) \\ \vdots \\ L_f^{\rho_r-1}h(x) \end{bmatrix} \triangleq \begin{bmatrix} \phi(x) \\ \hline \varphi(x) \end{bmatrix} \triangleq \begin{bmatrix} \varsigma \\ \hline \xi_e \end{bmatrix} \quad (5.2)$$

$$(5.3)$$

To prevent that $\dot{\varsigma}$ does not depend on the input u , the $\phi(x)$ functions are chosen such that

$$\frac{\partial \phi_i}{\partial x} g(x) = 0 \text{ for } 1 \leq i \leq n - \rho_r \quad (5.4)$$

Definition 5.1.3 (Normal Form System)[29]

The relation 5.2 will bring the system 5.1 into a normal form 5.5-5.7,

$$\dot{\varsigma} = f_0(\varsigma, \xi_e) \quad (5.5)$$

$$\dot{\xi}_e = A_c \xi_e + B_c \beta(x)[u - \alpha(x)] \quad (5.6)$$

$$y = C_c \xi_e \quad (5.7)$$

Where, $\varsigma \in \mathbb{R}^{\rho_r}$ is the vector of internal state variables, $\xi_e \in \mathbb{R}^{n-\rho_r}$ is the vector of external state variables, (A_c, B_c, C_c) are matrices in canonical form representation. The terms $\gamma(x)$, $\alpha(x)$ and the function $f_0(\varsigma, \xi_e)$ are defined as follows.

$$f_0(\varsigma, \xi_e) = \left. \frac{\partial \phi}{\partial x} f(x) \right|_{x=T^{-1}(z)} \quad (5.8)$$

$$\gamma(x) = L_g L_f^{\rho_r-1} \quad (5.9)$$

$$\alpha(x) = -\frac{L_f^{\rho_r} h(x)}{L_g L_f^{\rho_r-1} h(x)} \quad (5.10)$$

Definition 5.1.4 (Input-Output Linear State Feedback Control)[29]

The control law that will bring the external part of the normal form into a linear one is stated as follows

$$u = \alpha(x) + \beta^{-1}(x)\nu \quad (5.11)$$

The equations 5.5-5.7 result in the system

$$\dot{\varsigma} = f_0(\varsigma, \xi_e) \quad (5.12)$$

$$\dot{\xi}_e = A_c \xi_e + B_c \nu \quad (5.13)$$

$$y = C_c \xi_e \quad (5.14)$$

Definition 5.1.5 (Relative degree)[29]

Let ρ_r be the number of times the system 5.1 is continuously derived till the output y meets the input u , resulting in the following form

$$y^{(\rho_r)} = L_f^{\rho_r} h(x) + L_g L_f^{\rho_r-1} h(x)u \quad (5.15)$$

Thus, a nonlinear system of the form 5.1 has a relative degree ρ_r , $1 \leq \rho_r \leq n$ if

$$L_f^i h(x) = 0, \quad i = 1, 2, \dots, \rho_r - 2 \quad (5.16)$$

$$L_g L_f^i h(x) \neq 0, \quad i = \rho_r - 1 \quad (5.17)$$

The system 5.15 can be input-output linearisable by the following equation

$$u = \frac{1}{L_g L_f^{\rho_r - 1} h(x)} \left[-L_f^{\rho_r} h(x) + \nu \right] \quad (5.18)$$

Reducing the system 2.7 into the following linear form

$$y^{(\rho_r)} = \nu \quad (5.19)$$

In order to apply the theory stated in section 5.1 to the SLM, the state representation stated in equations 4.16 and 4.17 must be arranged in a general nonlinear form system as presented in definition 5.1.1 and seen in equation 5.20.

$$\begin{bmatrix} \dot{q} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} S(q)u \\ -H^{-1}f \end{bmatrix} + \begin{bmatrix} 0 \\ H^{-1}f \end{bmatrix} \tau \quad (5.20)$$

In the previous matrix expression the input u is defined as $[\eta, \dot{\beta}_1, \dot{\beta}_2]$, however the angular velocities $[\dot{\beta}_1, \dot{\beta}_2]$ are not taken into account in this model, because \dot{q} depends only on η then $u = \eta$ and $\dot{u} = \dot{\eta}$.

Moreover, a nonlinear feedback $\tau = F^\dagger[H\nu + f]$ is applied to the state representation bringing the system to the form of 5.1

$$\begin{aligned} \begin{bmatrix} \dot{q} \\ \dot{u} \end{bmatrix} &= \begin{bmatrix} S(q)\eta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \nu \\ y &= h(q) \end{aligned} \quad (5.21)$$

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \dot{u} \end{bmatrix}, \quad f(x) = \begin{bmatrix} S(q)u \\ 0 \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5.22)$$

In order to achieve input-output linearization, an analysis of the output equation $y = h(q)$ must be taken. In other words, the position vector $q = [x_c, y_c, \theta]$ is of interest then three output equations are formulated as stated in the matrix expression 5.23.

$$h(q) = \begin{bmatrix} h_1(q) \\ h_2(q) \\ h_3(q) \end{bmatrix} = y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix} \quad (5.23)$$

According to the definition 5.1.5, the output equation 5.23 must be derived till it finds the input $\dot{\eta}$. Expression 5.24 shows that the output y has been derived twice till it finds the input $\dot{\eta}$, thus having a relative degree $\rho_r = 2$.

$$\begin{aligned} \dot{y} &= Jh(q)S(q)\eta \\ \ddot{y} &= \dot{\Phi}(q)\eta + \Phi\dot{\eta} \end{aligned} \quad (5.24)$$

Where:

$$J = \begin{bmatrix} \frac{\partial q}{\partial x_c} \\ \frac{\partial q}{\partial y_c} \\ \frac{\partial q}{\partial \theta} \end{bmatrix}, \quad \Phi(q) = Jh(q)S(q) \quad (5.25)$$

J is the Jacobian. Since $\dot{q} = \dot{\xi} = R^T(\theta)\Sigma(\beta)\eta$ equation 5.24 can also be arranged as follows:

$$\begin{aligned}\dot{y} &= R^T(\theta)\Sigma(\beta)\eta \\ \ddot{y} &= \dot{R}^T(\theta)\Sigma(\beta)\eta + R^T(\theta)\dot{\Sigma}(\beta)\eta + R^T(\theta)\Sigma(\beta)\dot{\eta}\end{aligned}\quad (5.26)$$

According to definition 5.1.2 a state variable transformation vector which is a diffeomorphism as defined in [31] is stated as follows.

$$z = T(q) = [z_{1:6}]^T = [y_{1:3}, \dot{y}_{1:3}]^T = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} h(q) \\ - - - \\ L_f h(q) \end{bmatrix} = \begin{bmatrix} h(q) \\ - - - \\ R^T(\theta)\Sigma(\beta)\eta \end{bmatrix}\quad (5.27)$$

The system under the new state variable transformation vector $T(q)$ is characterized by the following dynamics:

$$\dot{z} = [\dot{z}_{1:6}]^T = \begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dot{\xi} \\ \frac{d}{dt}(R^T(\theta)\Sigma(\theta)\eta) \end{bmatrix} = A_c z + B_c \beta(z)[\nu - \alpha(z)]\quad (5.28)$$

Where:

$$A_c = \begin{bmatrix} 0_{3 \times 3} & \mathbb{I}_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, B_c = \begin{bmatrix} 0_{3 \times 3} \\ \mathbb{I}_{3 \times 3} \end{bmatrix}\quad (5.29)$$

$\beta(z)$ and $\alpha(z)$ are found by calculating $\frac{d}{dt}(R^T(\theta)\Sigma(\theta)\eta)$ as it is shown as follows:

$$\beta(z)[\nu - \alpha(z)] = \frac{d}{dt}(R^T \Sigma(\theta) \eta)$$

Where :

$$\beta(z) = R^T(\theta) [\Sigma(\beta) \quad S_g \eta]$$

$$\alpha(z) = -\beta^{-1}(z) \dot{R}^T(\theta) \Sigma(\theta) \eta$$

$$\hat{\nu} = [\dot{\eta} \quad \zeta_1 \quad \zeta_2]$$

$$S_g = \begin{bmatrix} (s_{11} + s_{12}) & (s_{13} + s_{14}) \\ (s_{21} + s_{22}) & (s_{23} + s_{24}) \\ (s_{31}) & (s_{32}) \end{bmatrix}$$

With :

$$s_{11} = -l_1 \cos(\beta_2) \sin(\beta_1 - \alpha_1)$$

$$s_{12} = l_2 \sin(\beta_1) \cos(\beta_2 - \alpha_2)$$

$$s_{13} = -l_1 \cos(\beta_2) \cos(\beta_1 - \alpha_1)$$

$$s_{14} = l_2 \cos(\beta_1) \sin(\beta_2 - \alpha_2)$$

$$s_{21} = -l_1 \sin(\beta_2) \cos(\beta_1 - \alpha_1)$$

$$s_{22} = -l_2 \cos(\beta_1) \cos(\beta_2 - \alpha_2)$$

$$s_{23} = l_1 \cos(\beta_2) \cos(\beta_1 - \alpha_1)$$

$$s_{24} = l_2 \sin(\beta_1) \sin(\beta_2 - \alpha_2)$$

$$s_{31} = \cos(\beta_1 - \beta_2)$$

$$s_{32} = -\cos(\beta_1 - \beta_2)$$

In the previous equations $\nu = [\dot{\eta} \quad \zeta_1 \quad \zeta_2]^T$ and $\hat{\nu} = [\dot{\eta} \quad \zeta_1 \quad \zeta_2]^T$, e.g. $\nu \neq \hat{\nu}$. This mean that the input to the model ν given by equation 5.21 is not the same as the input $\hat{\nu}$ given by the design control law. This problem can be solved by differentiating the last two terms of $\hat{\nu}$.

Moreover, choosing a state feedback control law of the form showed in equation 5.11, $\hat{\nu} = \alpha(z) + \beta^{-1}(z)\vartheta$ brings the system 5.28 into a linear one of the form presented in equation 5.30.

$$\dot{z} = A_c z + B_c \vartheta\quad (5.30)$$

This linearized system is a controllable one. And, this can be easily verified just looking at the matrices A_c and B_c which are in controllable canonical form. The vector ϑ which contains the accelerations from the linear controller has to be designed in order for the robot to follow a reference input r . According to the linear control theory [32] the control input ϑ that allows the system to follow the reference input r is stated in equation 5.31.

$$\vartheta = -Kz + (N_u + KN_z) \quad (5.31)$$

Where :

$$N_u = -B_c^{-1}A_cC_c^{-1}$$

$$N_x = C_c^{-1}$$

$$r = [x_c, y_c, \theta, \dot{x}_c, \dot{y}_c, \dot{\theta}]^T$$

The values of the matrix $K_{3 \times 6}$ can be designed according to linear control theory [32], for that the poles p of the linear system are placed on the left hand side of the real axis, e.g. $p_{1,1} = -3.0$, $p_{1,2} = -3.001$, $p_{1,3} = -3.002$, $p_{1,4} = -3.003$, $p_{1,5} = -3.004$, $p_{1,6} = -3.005$. Then, by means of a function in octave4 called `place()` that accepts the matrices A_c and B_c and the vector p as inputs can calculate the values of the matrix K , e.g. $K = \text{place}(A_c, B_c, p)$. Figure 5.1 shows the linearized system together with the reference control input r .

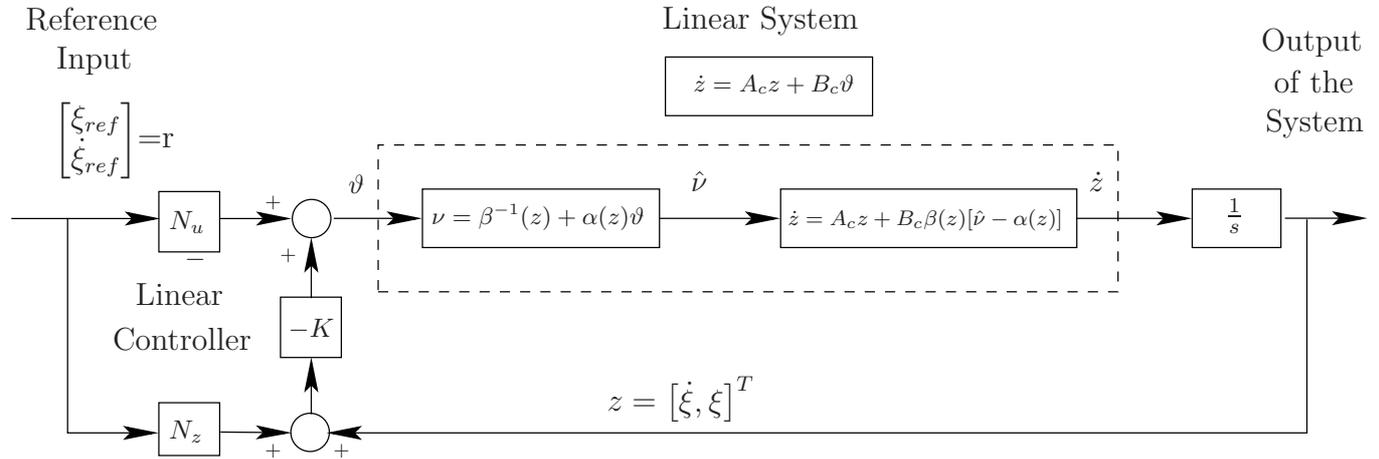


Figure 5.1: Shows the linear system together with the reference input

5.2 Control Simulation Results

Since β_3 and β_4 has to follow β_{c1} and β_{c2} respectively in order to meet the ICR, to this end β_3 and β_4 were deleted from the dynamic model. And, in order to verify the nonlinear control design together with the reference input, a reference trajectory was established as shown in expression 5.32.

$$r = \begin{bmatrix} \xi_{ref} \\ \dot{\xi}_{ref} \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ \theta \\ \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} t \\ e^{ct} - 1 \\ \arctan(ce^{ct}) \\ 1 \\ ce^{ct} \\ \frac{c^2 e^{ct}}{1+c^2 e^{2ct}} \end{bmatrix} \quad (5.32)$$

The system was simulated in Octave4 where the initial conditions of the simulation are chosen as: $x_c = 2.0$, $y_c = -0.5$, $\theta = 0.1$, $\beta_1 = 1$, $\beta_2 = 1$, $\eta = 0.2$. The results are shown in Figure 5.2 where 5.2(a) and 5.2(b) show the angles of the wheels 1 and 2 respectively, whereas 5.2(c) shows the angle θ and 5.2(d) shows the reference input $r = [\xi_{ref}, \dot{\xi}_{ref}]^T$ and the output of the system $z = [z, \dot{z}]^T = [\xi, \dot{\xi}]^T$. It can be seen that the robot being at initial position of $x_c = 2.0$ and $y_c = -0.5$ the controller tracks the reference input approximately at the position $x_c = 1.6$ and $y_c = 3.0$.

Figure 5.3 plots the position of the SML during the trajectory tracking and also shows the four steering wheel angles $\beta = [\beta_1, \beta_2, \beta_3, \beta_4]^T$ that follow the trajectory and also meet the ICR. During the trajectory the angles β_3 and β_4 follow the angles β_1 and β_2 in order to meet the ICR. The previous is achieved by equations 5.33 and 5.34, [31].

$$\beta_3 = \arctan\left(\frac{\cos(\beta_1) \sin(\beta_2)}{\sin(\beta_1 - \beta_2) + \cos(\beta_1) \cos(\beta_2)}\right) \quad (5.33)$$

$$\beta_4 = \arctan\left(\frac{\sin(\beta_1) \cos(\beta_2)}{\sin(\beta_1 - \beta_2) + \cos(\beta_1) \cos(\beta_2)}\right) \quad (5.34)$$

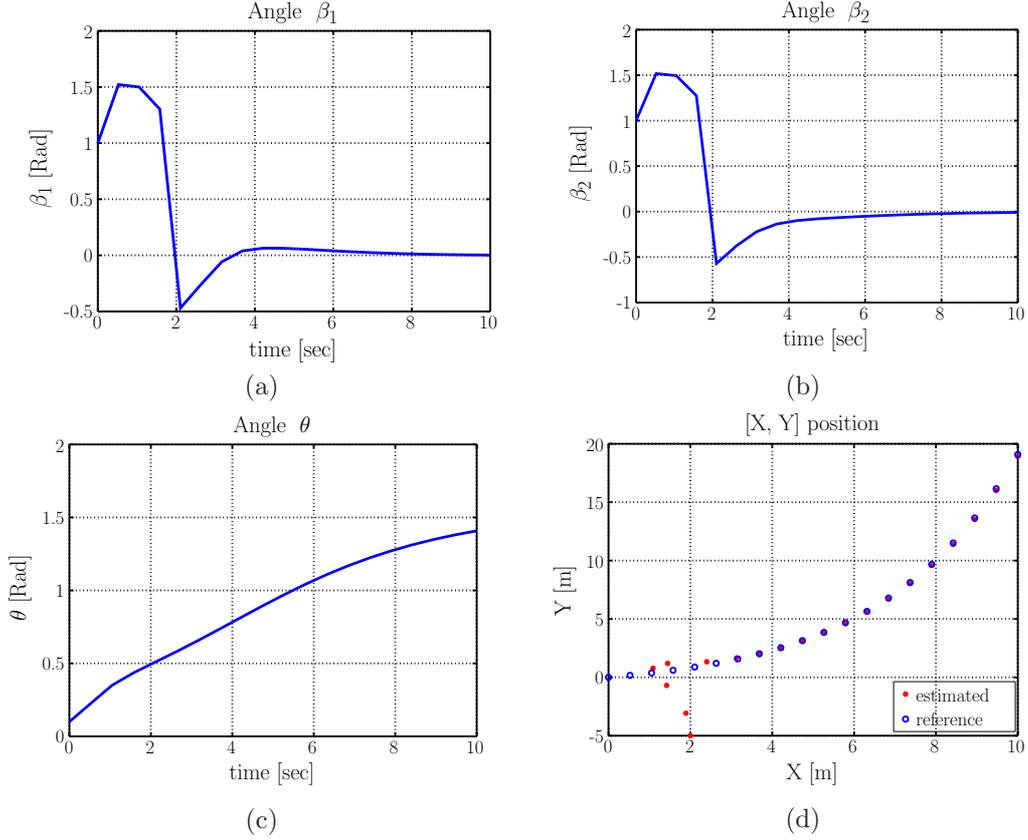


Figure 5.2: (a) Angle β_1 . (b) Angle β_2 . (c) Angle θ . (d) Position of the robot (x_c, y_c) and the reference input.

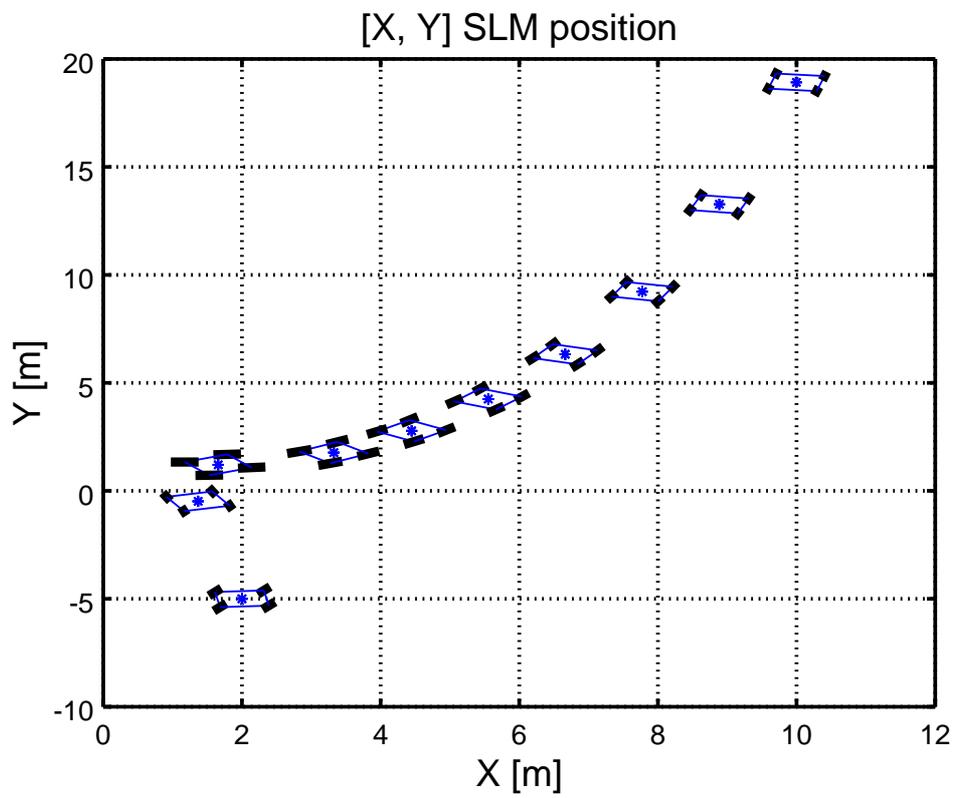


Figure 5.3: The Figure shows the SLM trajectory tracking position and also shows the four steering wheel angles $\beta_1, \beta_2, \beta_3, \beta_4$ that follow the trajectory and also meet the ICR.

Chapter 6

Motor Control

6.1 DC Motor model

To accurately model a DC motor, an electrical as well as a mechanical part are taken into consideration, [32].

6.1.1 Electrical Part

Figure 6.1 depicts the electrical part of a DC motor

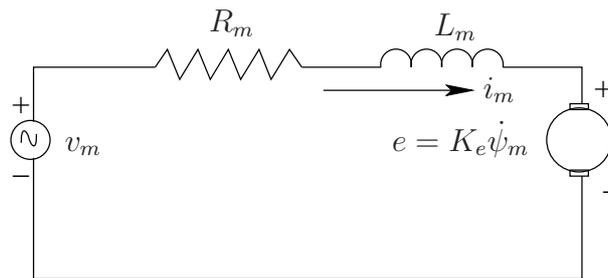


Figure 6.1: Motor electrical circuit

i_m [A], is the armature current, v_m [v], is the voltage applied to the motor, R_m [Ω], is the electrical resistance, L_m [H] is the electrical inductance, e [emf] electromotive force, K_e electric constant, $\dot{\psi}_m$ [$\frac{Rad}{sec}$] is the shaft's rotational velocity. The back emf, e , is related to the rotational velocity by the K_e .

Applying Kirchhoff voltage law $\sum_{k=1}^n v_k = 0$ to the electrical circuit in Figure 6.1 brings the electrical equation of the DC motor.

$$L_m \frac{di_m}{dt} + R_m i_m = v_m - K_e \dot{\psi}_m \quad (6.1)$$

6.1.2 Mechanical Part

Figure 6.2 depicts the mechanical part of a DC motor

T [$N \cdot m$] is the motor torque, ψ_m [Rad] is the shaft's angular position, b_m [$N \cdot m \cdot sec$] is the damping ratio of the mechanical system, J_m [$\frac{kg \cdot m^2}{sec^2}$] is the moment of inertia of the rotor, K_t [$\frac{N \cdot m}{A}$] is the torque constant. The motor torque, T , is related to the armature current, i_m , by a constant factor K_t .

Applying Newton's law to one dimensional rotational system ($M = I\alpha$), brings the mechanical equation of the DC motor, where M [$N \cdot m$] is the sum of all the moments about the center of mass, I [$K_g \cdot m^2$] is the body's moment of inertia about its center of mass, α [$\frac{Rad}{sec^2}$] is the angular acceleration of the body,

$$J_m \ddot{\psi}_m + b \dot{\psi}_m = K_t i_m \quad (6.2)$$

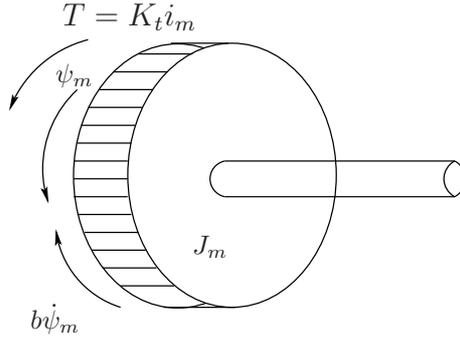


Figure 6.2: Motor mechanical part

6.1.3 Mathematical Model

Equations 6.1 and 6.2 represents the mathematical model of a DC motor

$$J_m \ddot{\psi}_m + b \dot{\psi}_m = K_t i_m \quad (6.3)$$

$$L_m \frac{di_m}{dt} + R_m i_m = v_m - K_e \dot{\psi}_m \quad (6.4)$$

The system can be arranged in state space representation.

$$\begin{bmatrix} \ddot{\psi}_m \\ \dot{i}_m \end{bmatrix} = \begin{bmatrix} -\frac{b}{J_m} & \frac{K_t}{J_m} \\ -\frac{K_e}{L_m} & -\frac{R_m}{L_m} \end{bmatrix} \begin{bmatrix} \dot{\psi}_m \\ i_m \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_m} \end{bmatrix} v_m \quad (6.5)$$

$$y = [1 \quad 0] \begin{bmatrix} \dot{\psi}_m \\ i_m \end{bmatrix} \quad (6.6)$$

Where (y) and $(\dot{\psi}_m, i_m)$ are the output and the internal states of the system respectively. However, for control purposes, the interest is to have a function of the form $G_{ol}(s) = \frac{\dot{\psi}_m(s)}{v_m(s)}$, an open loop transfer function in the s plane that relates the output $\dot{\psi}_m(s)$ to the input $v_m(s)$ is shown in equation 6.7.

$$G_{ol}(s) = \frac{\dot{\psi}_m(s)}{v_m(s)} = \frac{K_t}{(L_m s + R_m)(J_m s + b_m) + K_t K_e} \quad (6.7)$$

In SI units, K_t has the same value as K_e but with different units.

6.1.4 Control Design

In the control design, the choice of proper specifications depends on the application. In this case, the reference to the control loop is an angular velocity which is generated by a nonlinear system based on a trajectory. Figure 6.3 depicts the transfer function $G(s)$, the control law $D(s)$, the velocity reference input $\dot{\psi}$, the motor output velocity $\dot{\psi}_m$, the error $e = \dot{\psi} - \dot{\psi}_m$ and the control correction v_m .

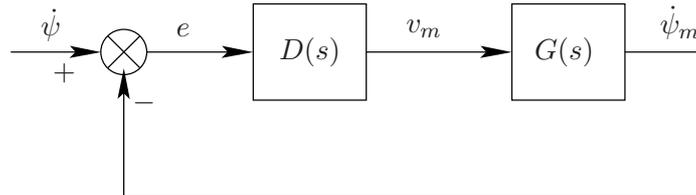


Figure 6.3: System control block

The described system $G(s)$ has two poles.

One pole is related to the electrical side and is very fast and mostly determined by the inductance L_m and the resistance R_m of the motor windings. It comes by letting $\psi_m = 0$ in equations 6.1 and 6.2

$$\frac{I_m(s)}{V_m(s)} = \frac{1}{L_m s + R_m} = \frac{1/R_m}{\tau_{ol_fast}} \quad (6.8)$$

Where $\tau_{ol_fast} = \frac{L_m}{R_m}$ is the open loop time electrical constant with the open loop pole $s_{ol_fast} = -\frac{1}{\tau_{ol_fast}} = -\frac{R_m}{L_m}$

The other pole is much slower and mostly determined by the mechanic part, and it can be depicted by letting $L_m = 0$ in equations 6.1 and 6.2

$$G_{ol}(s) = \frac{K_m}{J_m R_m s + (b_m R_m + K_m^2)} = \frac{K_m / (b_m R_m + K_m^2)}{\tau_{ol_slow} s + 1} \quad (6.9)$$

Where $\tau_{ol_slow} = \frac{J_m R_m}{J_m R_m + K_m^2}$ is the open loop time mechanical constant with the open loop pole $s_{ol_slow} = -\frac{J_m R_m}{J_m R_m + K_m^2}$.

One has to be aware that if the motor has some load with inertia this will change the slowest of the poles because it must exchange J_m with the total moment of inertia.

The former analysis was to emphasize that there is one pole related to the electrical side and one pole related to the mechanical side. In the design, the poles for the system $G(s)$ must be used.

$$s_{ol_slow_1} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (6.10)$$

$$s_{ol_fast_2} = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad (6.11)$$

$A = J_m L_m$, $B = -J_m R_m + L_m b_m$ and $C = b_m R_m + K_m^2$. There is no point in designing the closed loop system to be much faster than the slowest pole, since this would imply, that the control output would often be saturated. For instance, one could design a controller to have the slowest closed loop pole ($4 \rightarrow 5$) times faster than the slowest open loop pole, e.g. $s_{cl_slow} = 4 \times s_{ol_slow}$. This is indeed much slower than the electrical constant, such that the closed loop could have two close loop real poles, e.g. one slow and one fast.

For a close loop system with slow and fast real poles, the rise time t_r will be almost as for a first order system $G_{ol}(s) = \frac{1}{\tau s + 1}$ with output function $y(t) = 1 - \exp(-\frac{t}{\tau})$. Since the poles of the close loop system are chosen to be in the real axis, the system does not experiment any overshooting but it may experiment a steady state error.

In order to reduce steady state errors, a lag compensator $G_{lag}(s) = K_{lag} \frac{(s+z)}{(s+p)}$ can be used, where K_{lag} is the gain, z is the zero, p is the pole and ($z > p$) is the condition to be a lag compensator. However, if it is desired to have a closed loop with no steady state errors, a Proportional Integral (PI) control can be used, equation 6.12.

$$D(s) = \frac{K_p (T_i s + 1)}{T_i s} \quad (6.12)$$

If $p = 0$ in the lag compensator behaves like a PI controller or viceversa, meaning that the pole at zero cancels the steady state error. The addition of a pole at zero makes a third order system, however one can choose for instance the integral time constant (T_i) such that the zero of the controller cancels the slow pole of the motor, e.g. $s = -\frac{1}{T_i} = s_{ol_slow}$ making the system behave as a second order one. Then, the proportional constant (K_p) must be calculated according to the slow and fast close loop poles system. Algebraic manipulation of equation 6.12 can bring it into a lag compensation form, equation 6.13.

$$D(s) = K_p \frac{(s + \frac{1}{T_i})}{(s + 0)} \quad (6.13)$$

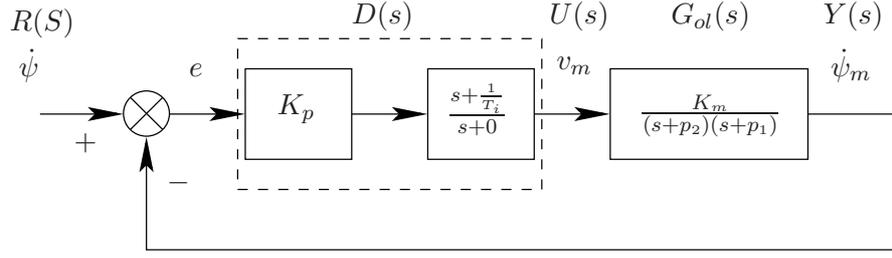


Figure 6.4: System control block with a PI acting as a lag compensator

The control loop system with PI acting as a lag compensator can be depicted in Figure 6.4. A close loop transfer function is obtained by a proper combination of equations 6.12 and 6.7.

$$H(s) = G_{cl} = \frac{G_{ol}D(s)}{1 + G_{ol}(s)D(s)} \quad (6.14)$$

6.2 Motor Control Simulation

This section shows the simulation results of a *PI* controller acting as a lag compensator that has been applied to a motor transfer function $G_{ol}(s)$.

The motor pololu (12V, 100:1 Gear Motor w/ 64 CPR Encoder) has been used to carry out the simulations and also the motor parameters has been calculated, appendix A. The obtained parameters are; $J_m = 0.044636$, $b_m = 0.011034$, $K_m = 0.95841$, $R_m = 2.4691$, and $L_m = 0.0027$. Figure 6.5(a) shows the slow and fast poles of the system $G_{ol}(s)$ which values are $s_{ol_slow} = -8.6614$ and $s_{ol_fast} = -906.07$.

Figure 6.5(b) depicts the step response of the open loop system. It can be seen there is a steady state error.

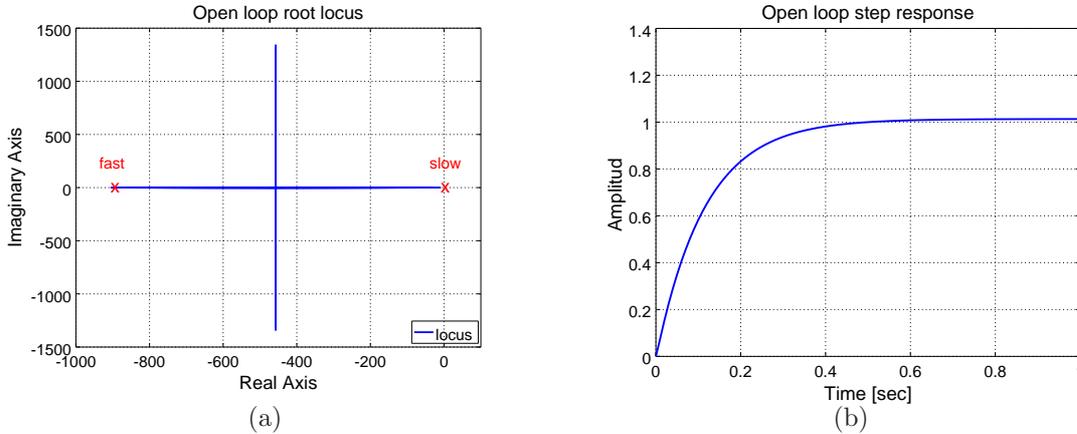


Figure 6.5: (a) Open loop poles. (b) Open loop step function.

The root locus in Figure 6.6(c) depicts the close loop poles and zeros of $G_{cl}(s)$. It can be seen that the system contains three poles and one zero. However, the entire system behaves as a second order one due to the fact that the integral constant TI in equation 6.12 has been chosen to cancel the slow open loop pole. Finally, a step response of the close loop system can be seen in Figure 6.6(d). Since the $G_{cl}(s)$ system behaves as a second order with one slow and one fast poles, the rise time t_r and the settling time t_s of the system can be obtained as follows

$$t_r = 2.2\tau_{cl_slow} \quad (6.15)$$

$$t_s = 3.9\tau_{cl_slow} \quad (6.16)$$

Where $\tau_{cl_slow} = -\frac{1}{s_{cl_slow}} = -\frac{1}{-25.2807} = 0.039556$ making $t_r = 0.087023$ sec and $t_s = 0.15427$ sec

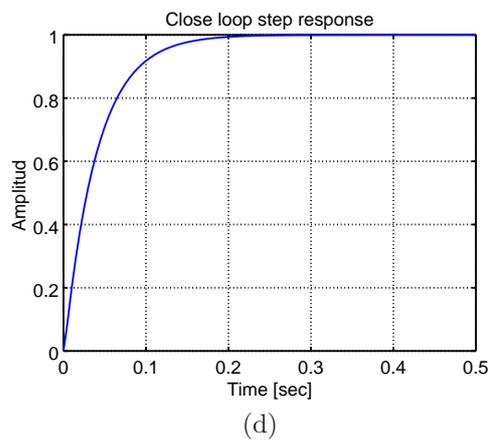
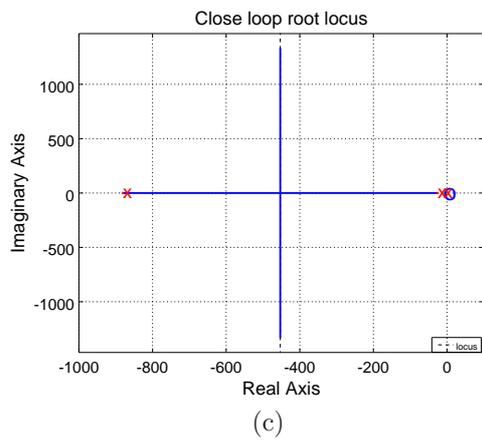


Figure 6.6: (c) Close loop poles. (d). Close loop step response.

Chapter 7

Digital Control

The equations of motion that govern dynamic systems as well as compensators must be digitized or discretized in order to be implemented in a digital computer. To this end, different tools for the discretization are available. For instance, nonlinear dynamic equations can be solved numerically, whereas compensators can be discretized by means of computer-aided control system design (*CACSD*), like Octave.

7.1 Nonlinear System discretization

The Euler's method [32] can be used to solve numerically the dynamic equations of motion that govern the nonlinearities of the SLM as shown in equation 7.1.

$$\dot{x} = \frac{x(k+1) - k(k)}{T} \quad (7.1)$$

where; $T = t_{k+1} - t_k$ is the sampling period, $x(k)$ is the value of x at time t_k , $x(k+1)$ is the value of x at time t_{k+1} .

To this end, a node was created in ROS that simulates the SLM nonlinear system. The results of the simulations can be depicted in Figure 7.1 where four trajectories can be seen; 1. the SLM trajectory (white line), 2. the reference trajectory (red line), 3. the angle β_1 (blue line) and 3. the angle β_2 (green line). And, the initial positions are; SLM $[-1.0, 0.5]$, reference trajectory $[0.0, 0.0]$, Angle β_1 $[0.3 \text{ Rad}]$ and angle β_2 $[0.3 \text{ Rad}]$. Then, at the start of the simulation the the SML starts driving towards the reference trajectory and in some point it is caught and continues following it. The angles tend to zero meaning that the wheel vertical planes are almost parallel to the base making the them follow the trajectory.

However, a more realistic path from a laboratory using a real map under ROS-RVIZ have been taken and stored in memory and then loaded with an Octave function for its analysis. The results of the nonlinear control simulation can be depicted in Figure 7.2(a-c). The angles β_1 and β_2 are shown in Figure 7.2(a)(b). It can be seen that these angles oscillate around 0 degrees, this means that they do not move far from the robot chassis, in other words, one can say that the nonlinear model generates the proper steering angles for the wheels to follow the trajectory path. Figure 7.2(c) depicts the angle θ which is the angle of the robot chassis frame with respect to the world frame. Finally the planned trajectory path from ROS-RVIZ is shown in Figure 7.2(d). This trajectory is colored as red and the nonlinear control that follows the path trajectory is colored as blue. It can easily be noticed how the control moves the chassis $[x_c, y_c]$ according to the planned path and as mentioned earlier it also generates the proper angles β_1 and β_2 .

The previous parameters β_1, β_2, θ and the nonlinear control path tracking $[x_c, y_c]$ which are depicted in Figure 7.2(a-c) have been taken as the inputs of an Octave function called "*animation2(x_c, y_c, \theta, \beta_1, \beta_2)*". The outcome of the simulated result is shown in Figure 7.3. In this figure, the position of the robot $[x_c, y_c]$ is represented as an asterisk during the tracking path, the chassis is represented as blue square which center is the position of the robot $[x_c, y_c]$, the wheels are represented as black rectangles which center are placed at each vertex of the chassis. It can be noticed that the orientation of the wheels, angles $\beta_1, \beta_2, \beta_3$ and β_4 correspond to the orientation of the tracking path and the perpendicular wheel lines shall meet a single ICR.

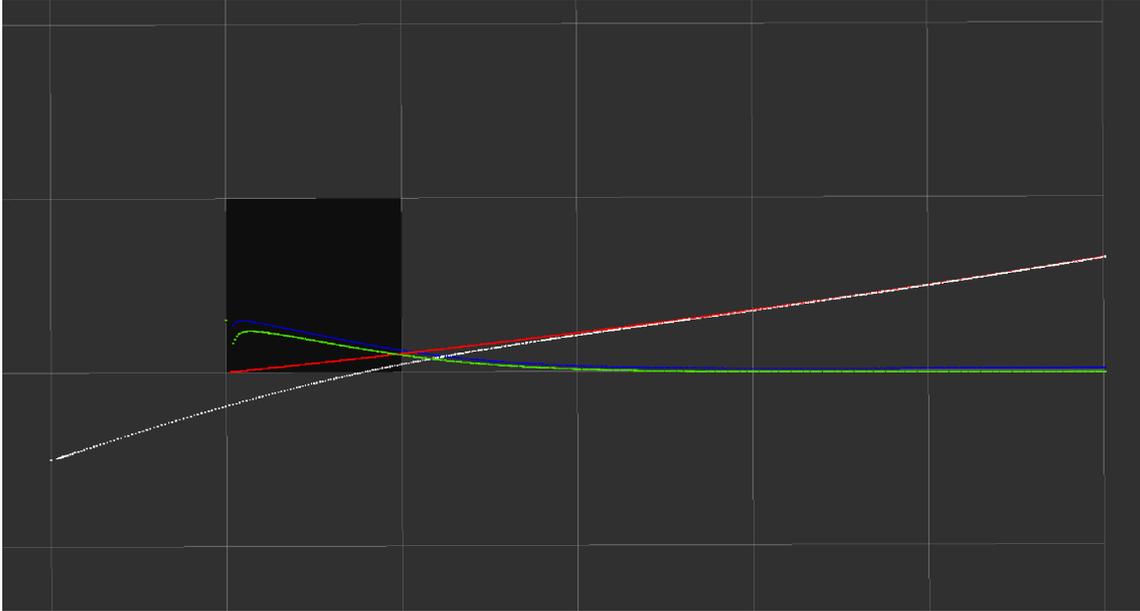


Figure 7.1: Four trajectories are shown; 1. The white line is the SLM trajectory. 2. The red line is the reference trajectory. 3. The blue line is the angle β_1 . 3. The green line is the angle β_2 .

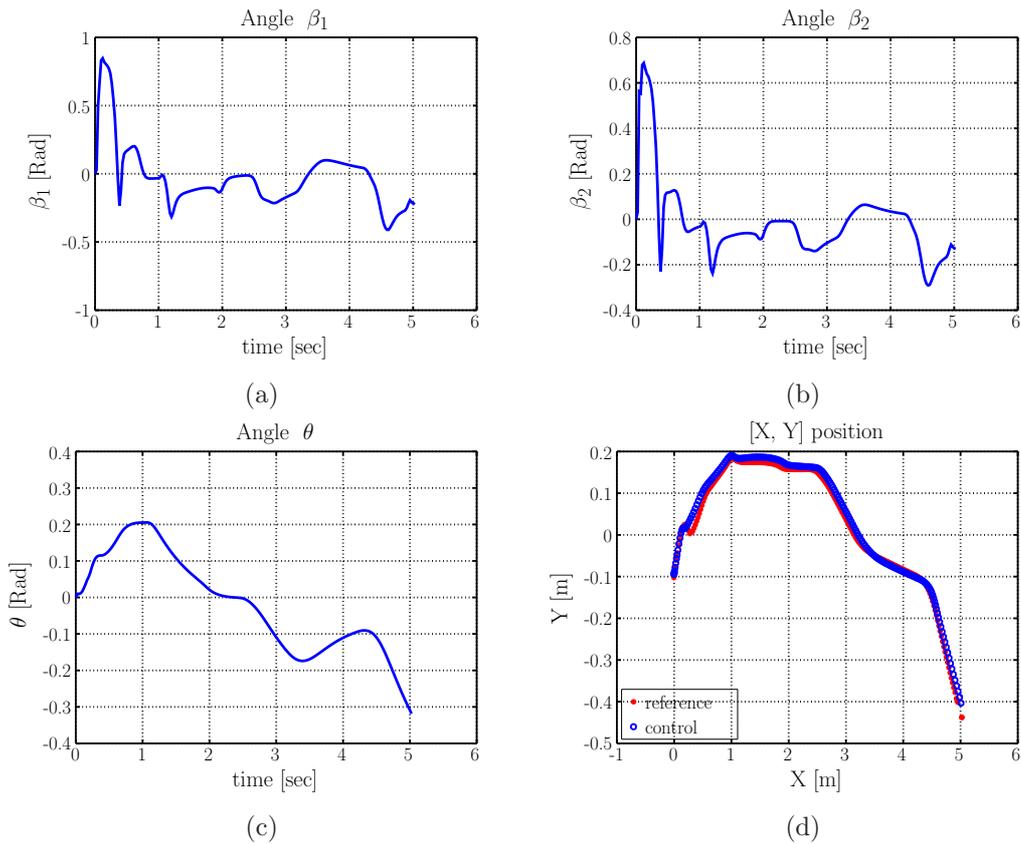


Figure 7.2: (a) Angle β_1 . (b) Angle β_2 . (c) Angle θ . (d) Control trajectory and the reference input.

7.2 DC Motor controller Digitization

Emulation is the discrete equivalent to a continuous closed loop DC motor transfer function $H(s) = G_{cl}(s)$. The digitization is achieved by means of Tustin's method, which mainly consists in approximate a continuous function using a trapezoidal integration, [32].

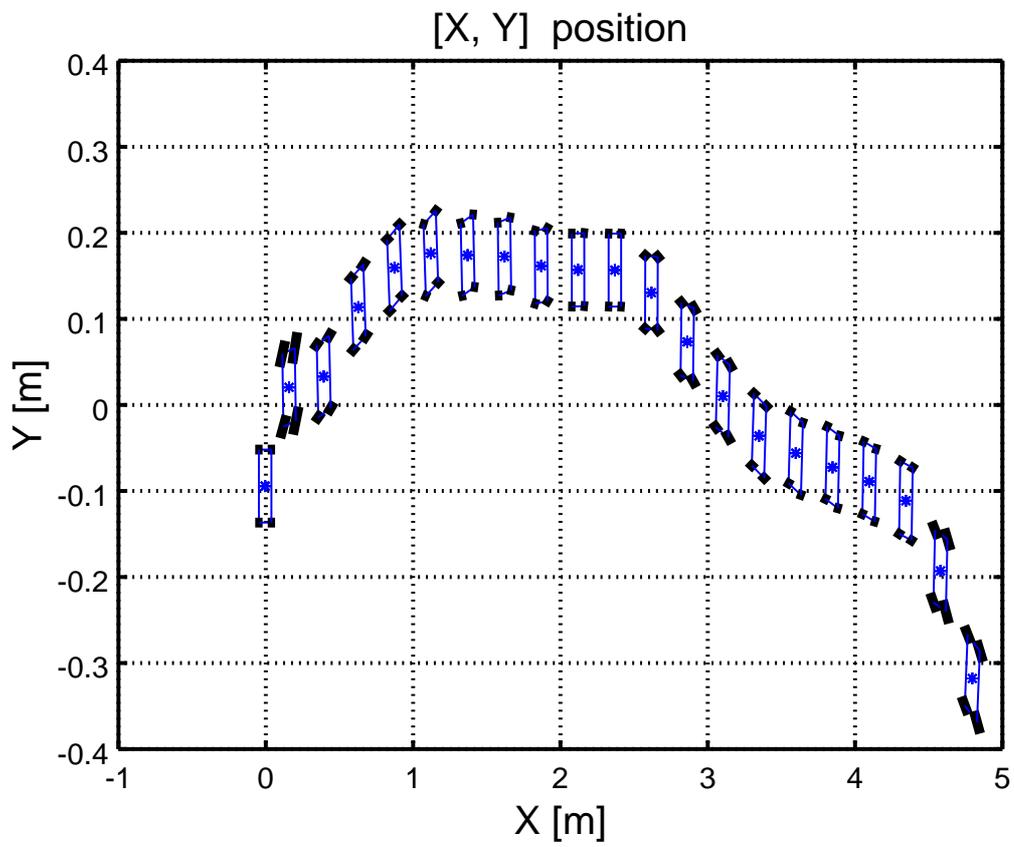


Figure 7.3: Shows the chassis and position of the center of gravity during the control trajectory. And, also shows the position of the wheels according to the trajectory.

Figure 7.4 shows a basic block diagram for a digital controller with a continuous real motor transfer function. The A/D block converts the continuous signal $y(t)$ from the system, in this case a DC motor. Then, y_k is compared with a reference signal r_k to produce an error signal e_k that is supplied to the difference equations, that have been discretized by Tustin's method. Hence, a correction signal u_k is produced over a D/A block to make the output $y(t)$ follow a reference input r_k .

An important factor in the digitization process is the selection of the sampling rate ω_n and the sample period T . ω_n is selected to be 20 times the bandwidth of the system $G_{ol}(s)$, whereas, T is selected by inverting ω_n .

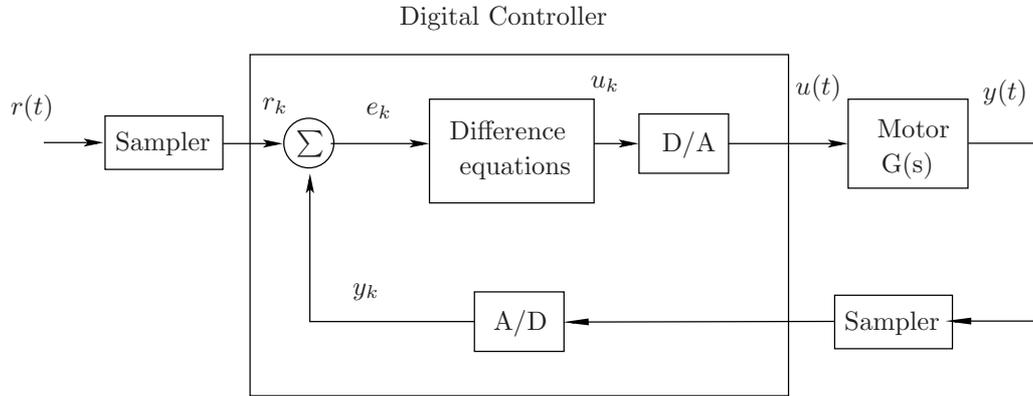


Figure 7.4: Digital block diagram controller

7.3 Digital Motor Control Simulation

Figure 7.5(a) shows the digitization of the open loop transfer function $G_{ol}(s)$. This is achieved by means of the octave function `c2d(system,T,'tustin')`, where `system` is the $G_{ol}(s)$ transfer function given by equation 6.9, `T` is the sampling period in seconds and `tustin` is digitization method. It can be seen that at the time of 0.15 [sec] the function gets an approximate amplitude of 0.7[Rad/sec] and also it experiments a steady state error which can not be seen in this plot. Figure 7.5(b) shows the digital step response of the close loop system $G_{cl}(s)$.

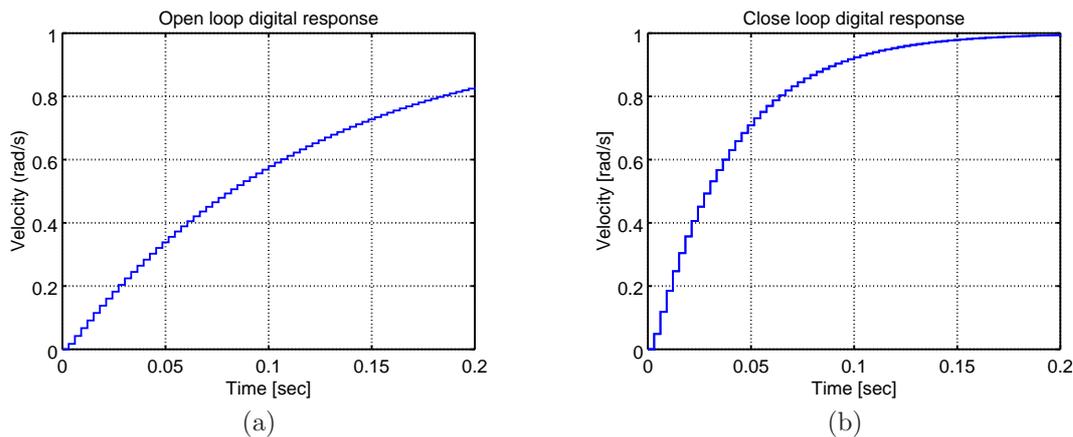


Figure 7.5: (a) shows the digitization of the step open loop transfer function $G_{ol}(s)$. (b) shows the digitization of the step close loop transfer function $G_{cl}(s)$.

Then, a comparison between digital and continues close loop step responses is shown in Figure 7.6(a). Moreover, a comparison between the motor and $G_{ol}(s)$ step responses is shown in Figure 7.6(b). It can be seen that this two responses do not match each other, the cause of this mismatch could be in the calculation of motor parameters, however for motor control purposes it is assumed that this two curves are good enough.

The digitization of the controller given by equation 6.13 with the aid of the octave function `c2d(system,T,'tustin')` produces the following control law in the z domain.

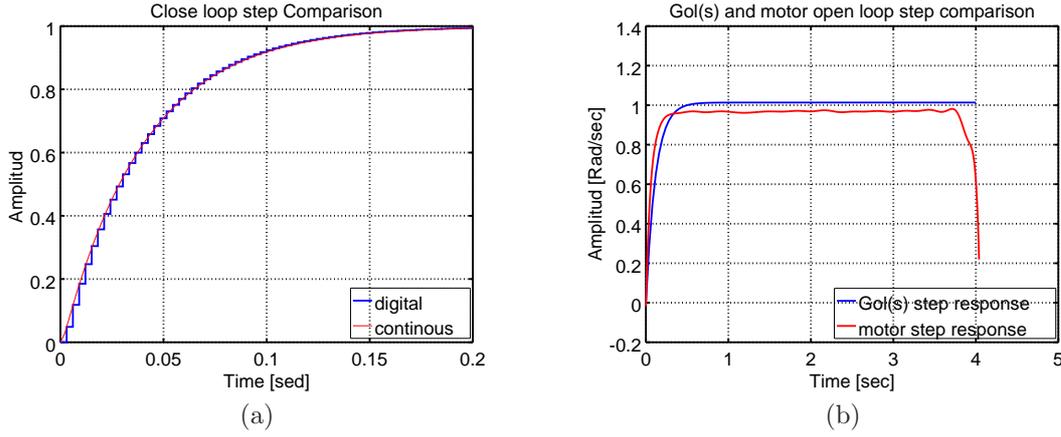


Figure 7.6: (a) Compares both digital and continues step responses of $Gol(s)$. (b) Compares both motor and $Gol(s)$ step responses.

$$D(z) = \frac{U(z)}{E(z)} = \frac{2.818 - 2.782z^{-1}}{1 - z^{-1}} \quad (7.2)$$

Then, isolating $U(z)$ in equation 7.2 and applying the inverse \mathcal{Z} transform $\mathcal{Z}[f(k-1)] = z^{-1}F(z)$, the difference equation as stated in equation 7.3 is obtained. Where, $U(k)$ is the value of the controller U at time t_k , $U(k-1)$ is the past value of U at time t_{k-1} , $e(k)$ is the error value of e at time t_k and $e(k-1)$ is the error value of e at time t_{k-1} .

$$u(k) = u(k-1) - 2.78e(k-1) + 2.818e(k) \quad (7.3)$$

The result of applying equation 7.3 to control the motor velocity to a given reference angular velocity that stem some nonlinear system is shown in Figure 7.7.

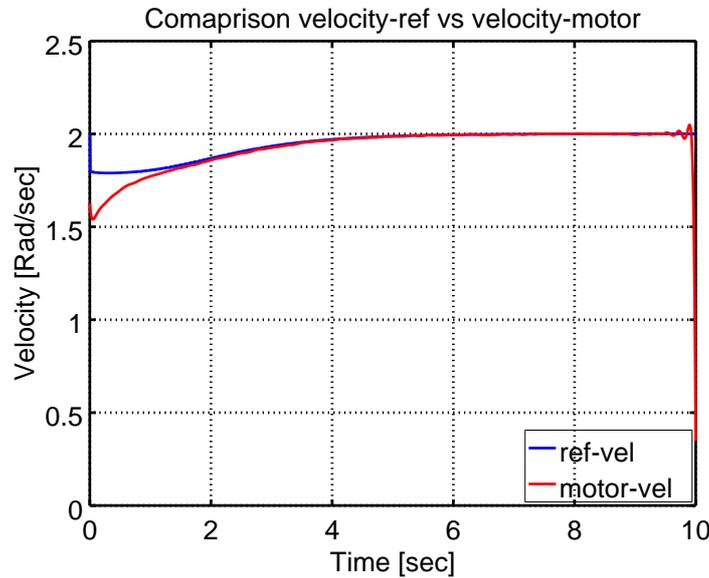


Figure 7.7: Shows the comparison between an angular velocity from a nonlinear system and the motor velocity control.

7.4 ROS-RVIZ System Simulation

The aim of this simulation is to show that the discretised nonlinear robot model, the nonlinear robot control as well as the linear control DC servos/motors make the robot to follow the nonlinear control tracked

path as it has been shown in Figure 7.2(c). To this end, the andromina robot as presented in Chapter 1 has been equipped with an Arduino Mega 2560 [33], Adafruit Motor/Stepper/Servo Shield for Arduino v2 Kit - v2.3 [34], Adafruit 16-Channel 12-bit PWM/Servo Shield [35] and a Hokuyo URG-04LX-UG01 laser scanner [36]. The equipped andromina robot can be depicted in Figure 7.8

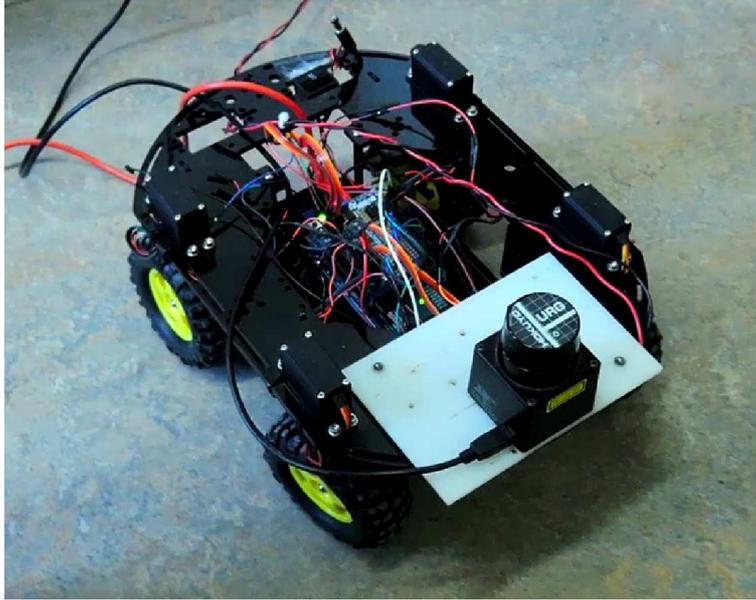


Figure 7.8: Shows the equipped andromina robot

The simulation process consists of the following steps:

- A node to store in memory a stable path using the default ROS path planning.
- A ROS launch file for the simulation that mainly consists of the following steps:
 - A node that runs the hokuyo laser ROS.
 - A node that loads the stable path.
 - A node that loads the map of the laboratory-office.
 - A node that runs a odometry based on the laser.
 - A node that runs the nonlinear model and control.
 - A node that connects ROS and arduino for motor/servo control.

The simulation of the test can be depicted in Figures 7.9 and 7.10. In the left upper window, the equipped andromina robot can be seen whereas the RVIZ graphical interface can be depicted in right down corner. The black area in the map represents a occupied area, the light gray represents the empty area and the dark gray that surrounds the occupied area is unknown.

Figure 7.9 shows the real as well as virtual andromina robot that have been placed close to the origin of the map frame where the light green line correspond to the laser scanner. Then, by selecting a goal point coordinate in RVIZ the coordinates of the stable path start loading at a frequency of 5Hz. After that, the nonlinear control algorithm starts tracking the path and generating the control commands for the motors/servos to follow the path as depicted in Figure 7.10. In this Figure the blue line corresponds to the stable path. The red line corresponds to the nonlinear control tracked path. The dark green line corresponds to the position of the laser frame with respect to the map frame that for some bug reason was shifted but it can be seen that corresponds to the path.

7.5 Conclusion

It is worth mention that in this simulation there is no feedback control. In other words, there is no localization algorithm that can estimate the robot position $[x_c, y_c, \theta]$ to be compared with the reference path in order that the controller can generate the right control commands. Instead, a kinematic model

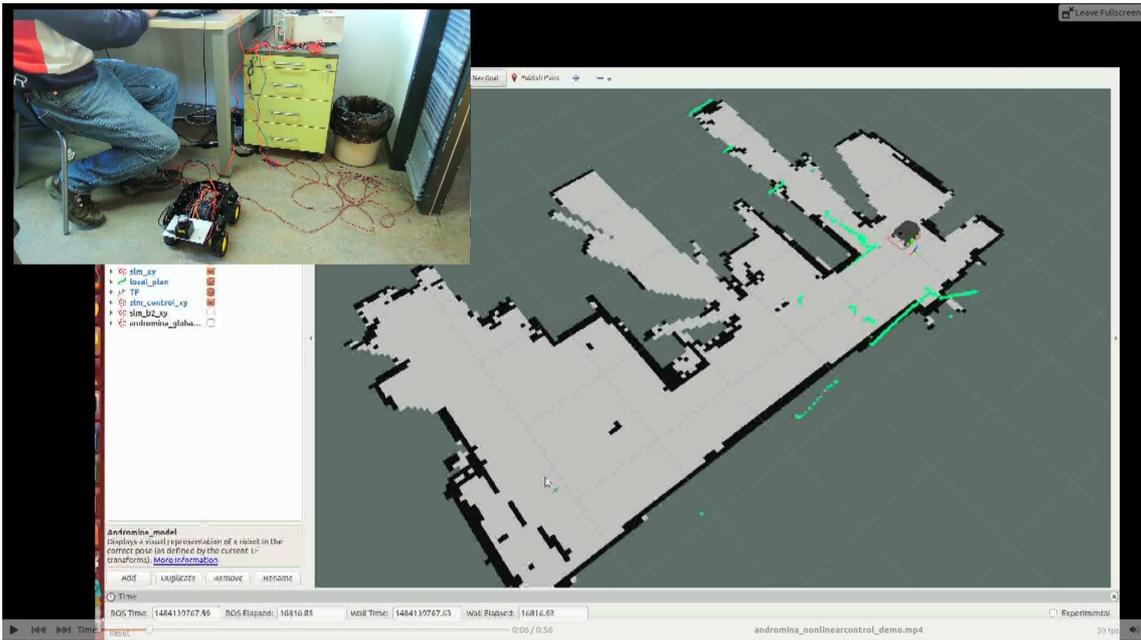


Figure 7.9: It shows the initial position of the andromina robot together with the laser scanned line (light green line).

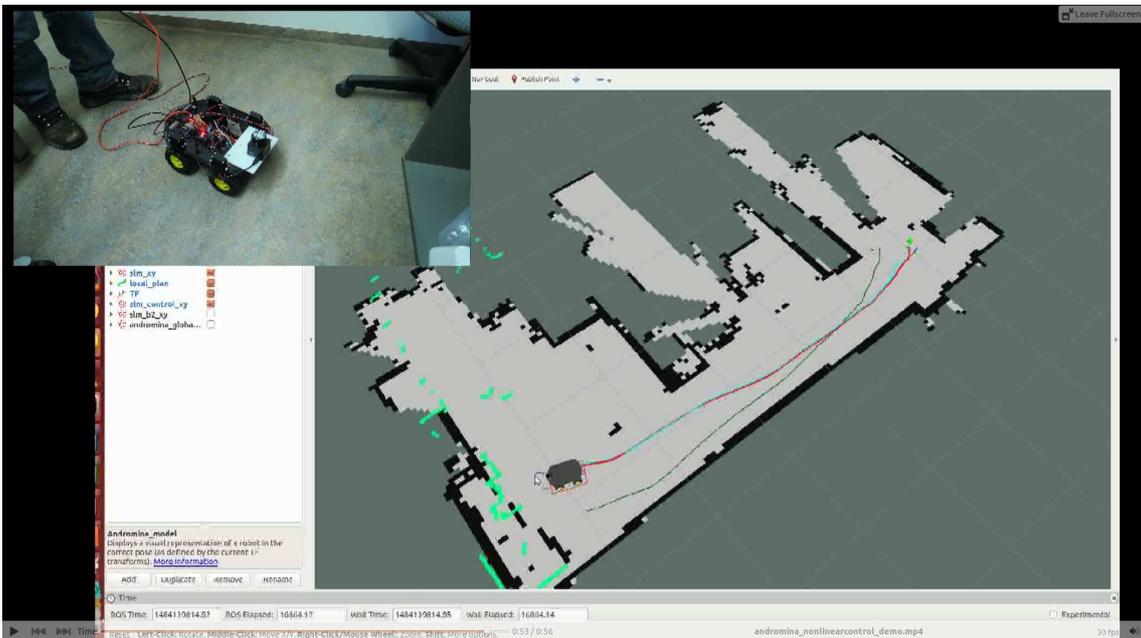


Figure 7.10: It shows the goal position of the andromina robot together with the stable path (sky blue line), the tracked path (red line), odometry position (dark green line).

equation 4.17 has been used for the simulation. The path used during the simulation is stable, this means that the control suffers from unbound nonlinearities. There is a necessity to research further under which conditions the control is unbounded and take them into account in the nonlinear control algorithm. It is believe that a proper mathematical analysis of the previous mentioned problem can lead to a publication of an article. Moreover, the control must be tested with a localization algorithm and perhaps with another path planner [37]. These issues will improve the ROS-pkg and is also believe an article for the ROS community can also be published.

Part III

Case Studies

Chapter 8

Case Study One: Move Control Correction and Optimization

This first case study of the SLM has to tackle two main issues:

- Avoid slippage conditions.
- Following cutted grass edge with no overlapping.

8.1 Slippage Conditions

Nonholonomic robots as the SLM are characterized by no-slippage constrains [22]. However, in many practical situations these conditions are violated and they need to be taken into account in the dynamics of the mobile robot [38]. Chapters 3, 4 and 5 deals with the mathematical model of the SLM and nonlinear control respectively under the constrain situations of rolling without slipping and no lateral movement.

Section 2.3 suggests ROS as a software tool to implement the SLM study cases, it has been seen in this section that NS uses the DWAP for free-collision path planning. This approach has mainly developed for synchro-drives. The synchro drive system is a two motor, three or four wheeled drive configuration where one motor rotates all wheels to produce motion and the other motor turns all wheels to change direction. The advantage of this approach is that it separates motors for translation and rotation making the control easier, however the the disadvantage of having separate motors is fairly complex.

Moreover, the SLM has a very different mechanism. It is a four steering wheel mobile robot, meaning that each wheel has two motors, one for translation and one for steering. It is suggested to implement the mathematical model and control stated in chapters 3, 4 and 5 respectively. The kinematic and dynamic models need to be simulated and the stability of the system need also to be tested. For a fast prototyping, Matlab can be used as a software tool, under success this model can be inserted in the NS as a node or as a plugin and then it can be simulated in Gazebo.

8.2 Following cutted grass edge with no overlapping

In order to improve the grass cutting process efficiency is necessary that SLM ditch cutted grass does not overlap with new ditch cutted grass. In other words, it is necessary that the SLM follows the ditch edge avoiding the overlapped process. According to [13, 9] this process can be achieved in two steps, ground and row detections. The following two sections mainly repeats what is stated in the previous articles and for further details refer to them.

8.2.1 Ground Detection

From a 3D point cloud the ground is detected. In order to achieved the previous process RANSAC is used to fit a Hessian plane equation data. Then the detected plane is refined by a least square fit. Depending on the field the detected ground can correspond to grass, soil or the canopy.

8.2.2 Row Detection

Once the ground has been detected, the point cloud is transformed into the ground plane coordinate frame. In this frame the z coordinate represents the height of the cutting or not cutting grass. Then RANSAC and a least square fit determines the row model.

8.2.3 Row Control

The row model that has been obtained in section 8.2.2 can be used as a reference input for the controller as it is shown in Figure 8.1.

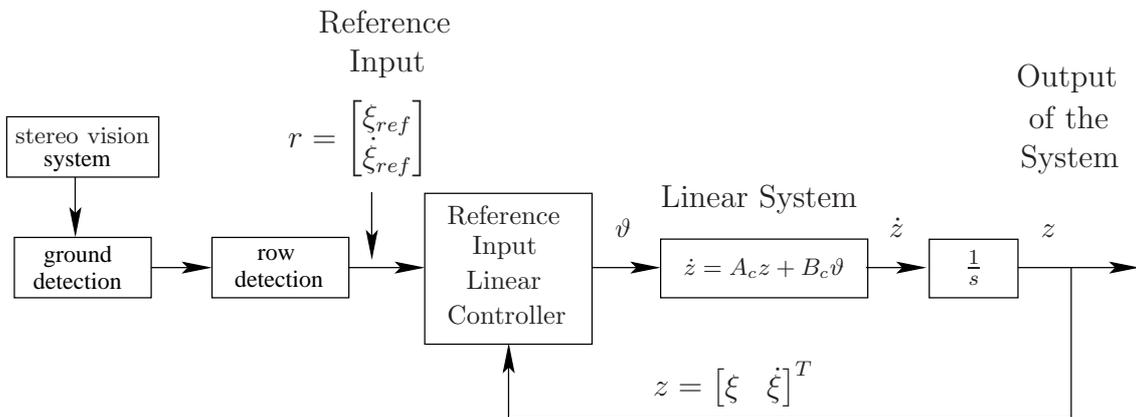


Figure 8.1: The vision system is used to detect 3D point cloud, then it is used for ground and row detections. The latter is then used as a reference input for the controller.

Chapter 9

Case Study Two: Coordination Control of Multi-Mobile Robot Systems

When dealing with coordination control of multiple mobile robot systems (CCMMRS) one has to do with the design and development of coordination control algorithms which meet different performance criteria according to the application, [39]. The most common approaches in CCMMR are: the master-slave/leader-follower, the virtual-structure, the behavioral based, and the motion-planning, approaches. In the following a brief review of those approaches are given and for further details refer to [39].

9.1 Master slave

In this approach one robot plays the role of a master and the rest of them as slaves as depicted in Figure 9.1. The master takes one task at hand and the slaves follow it as a group. However, in this coordination the information is unidirectional, meaning that the information flows from master to slaves and not vice versa. The disadvantages of this approach is that a failure in the master side will cause a failure of the whole group. And, one advantage is that even if the slaves have limited sensing and control capabilities, the group still can achieve its collective goal since the master directs the necessary movement to the slaves. Some classical control techniques to tackle the issue of CCMMRS are found in the literature. For instance, in [40] and [41] use feedback linearization control, and no linear control strategies as backstepping and sliding-mode control are used in [42] and [43] respectively.

9.2 Virtual structure

Figure 9.2 shows the virtual structure approach. The virtual center serves as a common reference where the rest of the robots generates their own trajectories based on that reference. In this mode all the robots communicate with each other and with the virtual center making the structure more robust against perturbations. Also, some work of virtual structure can be found in the literature. For instance, [44] and [45] equip each robot with a controller that enables formation keeping with other robots and simultaneously they track an individual reference trajectory. The formation keeping is achieved by the coupling between robots which require that all robots communicate with each other. The communication demand is reduced in [46] where the robots are only allowed to communicate only with the robots in its neighborhood. The reduced communication load is a very important property since the group should achieve high robustness at minimum communication cost.

9.3 Behavior-based

This formation control approach is based on robot input motion primitives which are weighted in order to compute robot's behavior [47]. The main advantage of this approach is its modularity, e.g. the capacity to decompose a complex task into modular subtasks. However, the main disadvantage is that it is hard

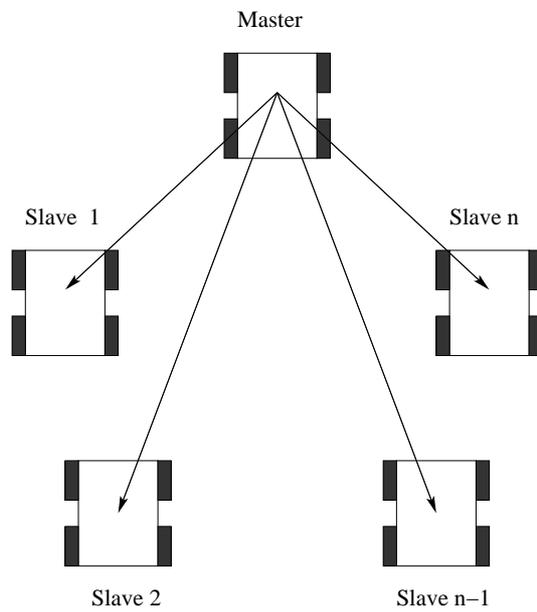


Figure 9.1: Shows the master-slave approach and the and the communication structure.

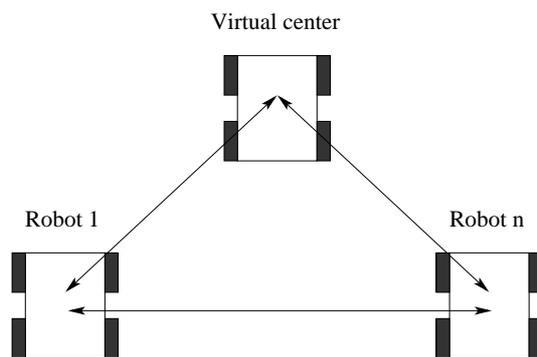


Figure 9.2: Shows the virtual-structure approach and the and the communication topology.

to analyze the dynamics mathematically. Then, as a consequence, the stability proof of the closed-loop system is difficult to analyze and the system cannot be predicted accurately.

9.4 Motion planning

As its name suggests, this approach is achieved by coordination of motion planning of the robots. For instance [48] shows how the robot coordination can be achieved via coordinated path-planning algorithms. Also, this high level planning approach is used in coordination of a group of robots such as Automated Guided Vehicles (AGV) inside manufacturing systems, see for example [49]. An optimization technique to design the coordination algorithms can be an advantage of this approach since it predicts the behavior of the system in advance. However, the resulting coordination algorithms work like a feedforward controller making the system not robust against perturbations.

An alternative to the previous mentioned approaches [39], [50] and [51] presents a hierarchical control approach to coordinate a group of mobile robots. The hierarchy consists of three main layers, namely high-level motion planner, low-level motion executor, and adjustable layer to accommodate the shifting of responsibilities. It also shows a coordination controller that is able to simultaneously track an individual trajectory and to keep a certain formation with other robots designed using dynamic feedback linearization. A performance comparison between high- and low-level coordination control for a group of mobile robots is also shown.

Chapter 10

Case Study Three: Living Person Detection

Chapter 11

Case Study four: Repeat Known Mission

Bibliography

- [1] M. Wasif, “Design and implementation of autonomous lawn-mower robot controller,” in *7th International Conference on Emerging Technologies (ICET)*. IEEE, 2011, pp. 1–5, ISBN: 978-1-4577-0769-8.
- [2] J. Smith, S. Campbell, and J. Morton, “Design and implementation of a control algorithm for an autonomous lawnmower,” in *Circuits and Systems, 2005. 48th Midwest Symposium on*, vol. 1. Covington, KY: IEEE, August 2005, pp. 456–459, ISBN: 0-7803-9197-7.
- [3] S. Bing-Min and L. Chun-Liang, “Design of an autonomous lawn mower with optimal route planning,” in *Industrial Technology, 2008. ICIT 2008. IEEE International Conference on*. Chengdu: IEEE, April 2008, pp. 1–6, ISBN: 978-1-4244-1705-6.
- [4] B. M. Taj and K. T. C. Timothy, “Design and modelling a prototype of a robotic lawn mower,” in *Information Technology, 2008. ITSIM 2008. International Symposium on*, vol. 4. Kuala Lumpur, Malaysia: IEEE, August 2008, pp. 1–5, ISBN: 978-1-4244-2327-9.
- [5] Spider. (2015) Mini, ILD01, ILD02. [Online]. Available: <http://www.slope-mower.com/products/>
- [6] Andromina. (2016) Andromina v.1.2. [Online]. Available: <http://androminarobot.blogspot.cz/search/label/Arduino>
- [7] Poyint-Gray. Cameleon3. [Online]. Available: <https://www.ptgrey.com/chameleon3-13-mp-mono-usb3-vision>
- [8] ——. (2012, june) Bumblebee2. [Online]. Available: <https://www.ptgrey.com/>
- [9] U. Weiss and P. Biber, “Plant detection and mapping for agricultural robots using a 3d lidar sensor,” *Robot. Auton. Syst.*, vol. 59, no. 5, pp. 265–273, May 2011, ISSN: 0921-8890. [Online]. Available: <http://dx.doi.org/10.1016/j.robot.2011.02.011>
- [10] Nerian-Vision-Technologies. (2016) SP1. [Online]. Available: <http://nerian.com/products/sp1-stereo-vision/>
- [11] DUO3D. (2016) Duo-MLX. [Online]. Available: <https://duo3d.com/product/duo-minilx-lv1>
- [12] Nipon-Signal. (2009, Jan.) FX6. [Online]. Available: <http://www.signal.co.jp/>
- [13] P. Biber, U. Weiss, M. Dorna, and A. Albert, “Navigation system of the autonomous agricultural robot bonirob. in workshop on,” in *Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production*, Vilamoura, Portugal, october 2012.
- [14] Nipon-Signal. FX8. [Online]. Available: http://www.signal.co.jp/vbc/mems_e/sensor/
- [15] RTK-GPS. Indiegogo. [Online]. Available: <https://www.indiegogo.com/projects/reach-first-affordable-rtk-gps-receiver#/>
- [16] Mouser-Electronics. (2016) Omron DT6. [Online]. Available: <http://eu.mouser.com/new/omronelectronics/omron-d6t/>
- [17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, 2009.
- [18] D. Fox, W. Burgard, D. Frank, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” in *in proc. of the national conference on artificial intelligence, AAAI*, 1999, pp. 343–349.

- [19] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *Robotics & Automation Magazine, IEEE*, vol. 4, no. 1, pp. 23–33, 1997. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.502>
- [20] ROS. (2016, Jan.) rtabmap_ros. [Online]. Available: http://wiki.ros.org/rtabmap_ros
- [21] M. Labbe and F. Michaud, “Online global loop closure detection for large-scale multi-session graph-based slam,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, Chicago, IL, USA, september 2014, pp. 2661–2666.
- [22] C. Guy, B. Georges, and D. A.-n. Brigitte, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, pp. 47–62, 1996.
- [23] S. Nilanjan, Y. Xiaoping, and K. Vijay, “Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robots,” Department of Computer & Information Science (CIS), University of Pennsylvania, Tech. Rep., 1992.
- [24] K. R. Symon, *Mechanics*, ser. Addison-Wesley World student series. Addison-Wesley Publishing Company, 1971, ISBN: 9780201073928. [Online]. Available: https://books.google.cz/books?id=JVk_AudwNtkC
- [25] P. Coelho and U. N. Nunes, “Lie algebra application to mobile robot control: a tutorial,” *Robotica*, vol. 21, pp. 483–493, 2003.
- [26] A. M. Bloch, *Nonholonomic Mechanics and Control*. Addison-Wesley Publishing Company, 2000, ISBN: 0-387-95535-6.
- [27] H. Goldstain, *Nonholonomic Mechanics and Control*. Addison-Wesley, 1980, ISBN: 0-201-02918-9.
- [28] H. Nijmeijer and v. d. A. Schaft, *Nonholonomic Mechanics and Control*. Springer-Verlag, 1990, ISBN: 978-1-4757-2102-7.
- [29] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002, ISBN: 0-13-067389-7.
- [30] H. J. Marquez, *Nonholonomic Control Systems Analysis and Design*. Wiley, 2003, ISBN: 0-471-42799-3.
- [31] P. Andersen, S. T. Pedersen, and J. Dimon, “Robust feedback linearization-based control design for a wheeled mobile robot.” Proc. of the 6th International Symposium on Advanced Vehicle Control, 2002.
- [32] G. F. Franklin and J. D. Powell, *Feedback Control of Dynamic Systems, third edition*. Addison-Wesley, 1994, ISBN: 0-201-53487-8.
- [33] Arduino. Arduino Mega 2560. [Online]. Available: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>
- [34] Adafruit. Adafruit Motor Shield. [Online]. Available: <https://www.adafruit.com/product/1438>
- [35] ——. Adafruit Servo Shield. [Online]. Available: <https://www.adafruit.com/product/1411>
- [36] Hokuyo. URG-04LX-UG01. [Online]. Available: https://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html
- [37] A. Valero-Gomez, J. V. Gómez, S. Garrido, and L. Moreno, “The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories,” *IEEE Robot. Automat. Mag.*, vol. 20, no. 4, pp. 111–120, 2013.
- [38] N. Sidek and N. Sarkar, “Dynamic modeling and control of nonholonomic mobile robot with lateral slip,” in *Proceedings of the 7th WSEAS International Conference on SIGNAL PROCESSING, ROBOTICS and AUTOMATION (ISPRA '08)*, WSEAS, Ed. WSEAS, february 2008, pp. 66–74, ISBN: 978-960-6766-44-2.
- [39] S. Adinandra, “Hierarchical coordination control of mobile robots,” Ph.D. dissertation, Technische Universiteit Eindhoven, Eindhoven, 7 2012, ISBN = 978-90-386-3204-9.

- [40] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots." *IEEE T. Robotics and Automation*, vol. 17, no. 6, pp. 905–908, 2001. [Online]. Available: <http://dblp.uni-trier.de/db/journals/trob/trob17.html#DesaiOK01>
- [41] R. Fierro, P. Song, A. Das, and V. Kumar, "Cooperative control of robot formations," 2001.
- [42] X. Li, J. Xiao, and Z. Cai, "Backstepping based multiple mobile robots formation control," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, september 2005, pp. 887–892, ISBN = 0-7803-8912-3.
- [43] M. Defoort, T. Floquet, A. M. Kksy, and W. Perruquetti, "Sliding-mode formation control for cooperative autonomous mobile robots." vol. 55, no. 11, pp. 3944–3953, 2008. [Online]. Available: <http://dblp.uni-trier.de/db/journals/tie/tie55.html#DefoortFKP08>
- [44] H. A. Thijs, v. d. Broek, v. d. N. Wouw, and H. Nijmeijer, "Formation control of unicycle mobile robots: a virtual structure approach," in *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*. Shanghai, P.R. China,,: IEEE, December 2009, pp. 8328–8333.
- [45] D. Kostic, S. Adinandra, J. Caarls, and H. Nijmeijer, "Collision-free motion coordination of unicycle multi-agent systems," in *American Control Conference (ACC)*. Baltimore, MD: IEEE, June-July 2010, pp. 3186 – 3191, ISBN = 978-1-4244-7426-4.
- [46] A. Sadowska, T. v. d. Broek, H. Huijberts, and v. d. W. Nathan, "A virtual structure approach to formation control of unicycle mobile robots using mutual coupling," *International Journal of Control*, vol. 84, no. 11, pp. 1886–1902, November 2011.
- [47] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 06, pp. 926–939, December 1998.
- [48] S. Petr and H. O. Mark, "Coordinated path planning for multiple robots," *Robotics and Autonomous Systems*, vol. 23, pp. 125–152, 1998.
- [49] E. K. Xidias and P. N. Azariadis, "Mission design for a group of autonomous guided vehicles," *Robot. Auton. Syst.*, vol. 59, no. 1, pp. 34–43, January 2011, ISSN: 0921-8890. [Online]. Available: <http://dx.doi.org/10.1016/j.robot.2010.10.003>
- [50] A. Sisdarmanto, K. Dragan, C. Jurjen, and N. Henk, "Towards a flexible transportation in distribution centers low-level motion control approach," in *Proceedings of the 8th International Conference on Informatics in Control, Automation and Robotics*. SCITEPRESS (Science and Technology Publications, Lda.), 2011, pp. 155–160, ISBN = 978-989-8425-75-1.
- [51] A. Sisdarmanto, C. Jurjen, K. Dragan, and N. Henk, "Performance of high-level and low-level control for coordination of mobile robots," in *Proceedings of the 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2010, pp. 63–71.

ACKNOWLEDGMENT

I would like to thank the all mighty God for allow me to contribute in this project according to his will. And also my gratitude goes to FIT and specially to VÍta Beran for giving me the opportunity to participate in this project.

Appendix A

Motor Parameters

The electrical and mechanical equations 6.1 and 6.2 respectively are stated here again.

$$L_m \frac{di_m}{dt} + R_m i_m = v_m - K_e \dot{\psi}_m \quad (\text{A.1})$$

$$J_m \ddot{\psi}_m + b \dot{\psi}_m = K_t i_m = \tau \quad (\text{A.2})$$

A.1 Electrical Resistance R_m

The electrical resistance R_m can be calculated from the electrical equation A.1. When the DC motor is in steady state, the term $\frac{di}{dt}$ becomes zero. Moreover, holding the motor shaft while applying voltage v_m the term $K_e \dot{\psi}$ also becomes zero. The previous actions leads equation A.1 to the form $R_m = \frac{v_m}{i_m}$. Then by measuring the applied voltage and the current the R_m can be calculated. Table A.1 shows different applied voltages V_m [volts] and its respective current i_m [A]. Figure A.1 shows a line plot of the resistance R_m and by taken the slope the resistance has a value of $R_m = 2.4691 \Omega$.

V_m [volts]	i_m [A]
1	0.28
2	0.58
3	1.10
4	1.5
5	1.9

Table A.1: Applied voltage and measured current to the motor.

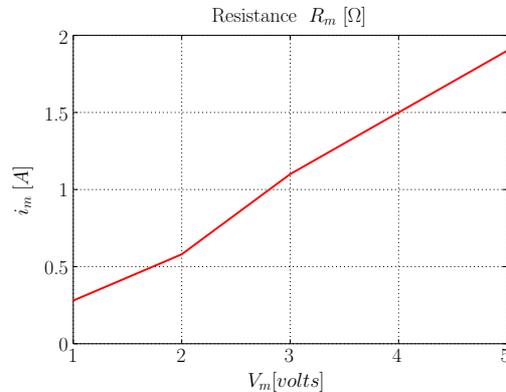


Figure A.1: Shows the plot of the applied voltage and measured current to the motor.

A.2 Motor Torque K_t and Electromotive force constants K_e

As mentioned earlier, in steady state the term $\frac{di}{dt}$ in the electrical equation A.1 becomes zero. To this end, different voltages have been applied to the motor and the corresponding current and angular velocity have been measured as depicted in Table A.2.

V_m [volts]	i_m [A]	$\dot{\psi}_m$ [Rad/sec]
2	0.16	1.42
3	0.20	2.35
4	0.22	3.34
5	0.20	4.34
6	0.21	5.38
7	0.22	6.40
8	0.23	7.50

Table A.2: Shows the applied voltage to the motor and the measured current and angular velocity.

Applying the values from Table A.2 to the equation A.3, K_e is obtained with a value of $K_e = 0.95841$. In the international system of units (SI) K_t and K_e has the same value but with different units, so for numerical calculations K_t and K_e is referred as K_m .

$$K_e = \frac{V_m - R_m i}{\dot{\psi}_m} \quad (\text{A.3})$$

A.3 Friction Coefficient b_m

The term $J_m \ddot{\psi}_m$ in the mechanical equation A.2 becomes zero when the motor is in steady state, then b_m can be calculated from the following linear equation A.4 which slope is b_m . Taking i_m and $\dot{\psi}_m$ from Table A.2 and the value of K_m from section A.2 into the equation A.4, the plot in Figure A.4 is produced. Then, calculation of the slope gives $b_m = 0.011034$.

$$b_m \dot{\psi}_m = K_m i_m \quad (\text{A.4})$$

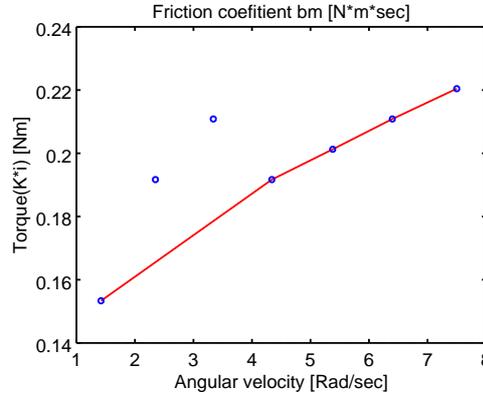


Figure A.2

A.4 Inertia of the Rotor J_m

This section will be added later :) :)

A.5 Electrical Inductance L_m

This section will be added later :) :)