Differential mobile robot based wheelchair

Plascencia, Alfredo Chavez; Dremstrup, Kim

Publication date:
2011

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Plascencia, A. C., & Dremstrup, K. (2011). *Differential mobile robot based wheelchair*. Department of Health Science and Technology. Aalborg University. Report / Aalborg University, Department of Health Science and Technology No. 2011:1

AALBORG UNIVERSITY

Department of
Health Science and Technology

Alfredo Chávez Plascencia and Kim Dremstrup

Differential Mobile Robot based Wheelchair

Report *2011:1*

Differential Mobile Robot based Wheelchair

Alfredo Chávez Plascencia and Kim Dremstrup
Center for Sensory-Motor Interaction (SMI),
Department of Health Science and Technology

Aalborg University, Denmark.

October 9, 2011

[1] Contact: Alfredo Chávez and Kim Dremstrup, e-mail: acp@iha.dk, kdn@hst.aau.dk

ISBN 978-87-7094-098-6

Contents

Part I

Preliminaries

Chapter 1

Introduction

To assist mobility of disable persons, several electrical powered wheelchairs are available on the market [1]. Many of them allow the use of different control modules to suit the needs of a high variety of users with different disabilities. An approach to assist mobility of disabled persons is via autonomous wheelchairs. Typically, these consists of either a standard power wheelchair where a computer and a collection of sensors have been added, or a mobile robot base to which a seat has been attached. Whatever is the case, the main idea is to assure a collision-free travel, e.g. avoiding obstacles and passing through doorways. To this end, the user can input the coordinates of the destination using the alternative control interface, e.g. clicking a point on the map built by the sensors, while the wheelchair performs localization and path planning tasks to autonomously transport the user to the desired location.

In this work, a differential drive mobile robot with a laser sensor mounted on the top and two encoders in each wheel has been built to implement a high-autonomy level wheelchair type control system. Autonomy is achieved by implementing modules of sensor fusion, map making, localization, path planing and control.

Different methods to implement the modules were reviewed and analyzed. For instance, Bayesian is a method that can be used to fuse and construct a probabilistic map of the environment into an occupancy grid, based on sensor data readings [64, 65]. Path planning strategy can be achieved by the solution of Laplace's equation to harmonic functions, [12, 13, 14, 15, 16]. Kullback Leibler Divergence (KLD) sampling, which is a variant of Montecarlo Localization (MCL) in the sense that adapts the number of samples overtime, is used for localization, [33]. Furthermore, for systems with holonomic as well as nonholonomic constraints and not fully state feedback linearizable, input-output state feedback control can be used, [48, 50, 58].

It is shown in simulation that the differential drive wheelchair prototype performs the map-making, localization, path planning and control to autonomously move from an actual localization to a desired location chosen by the point selection block, avoiding any obstacles in its way.

This report is organized as follows: part I contains the introduction and the description of the system. Part II describes the theoretical background used to implement the modules mentioned previously. And, finally, part III shows the simulation results of each module as well as the system respectively.

Chapter 2

System Description

The system in turn is a small prototype of an autonomous wheelchair, which is presented in figure 2.1 and mainly consists of the following:

- EPIA M10000 Mini ITX Mainboard + 1 GHz CPU.
- Two DC motors.
- Laser.
- Motor interface.

2.1 Mainboard

EPIA M10000 Mini ITX Mainboard + 1 GHz CPU board was chosen mainly for two reasons: it's economical and it's compatibile to linux system. The main characteristics of the board are presented in table 2.2.

Table 2.1: EPIA M10000 Mini ITX Mainboard

Model name	Epia board
Processor	1.0GHz VIA
System Memory	1GB memory size
Onboard I/O Connectors	3 x USB 2.0 ports 2 x IEEE1394 connectors 1 x Serial port 1 x Parallel port
Onboard	LAN 1 x 10/100 Mbps Ethernet
Expansion Slot	1 x PCI slot
Chipset	VIA CLE266 North Bridge
Dimensions	17×17

The board can be seen in figure 2.2

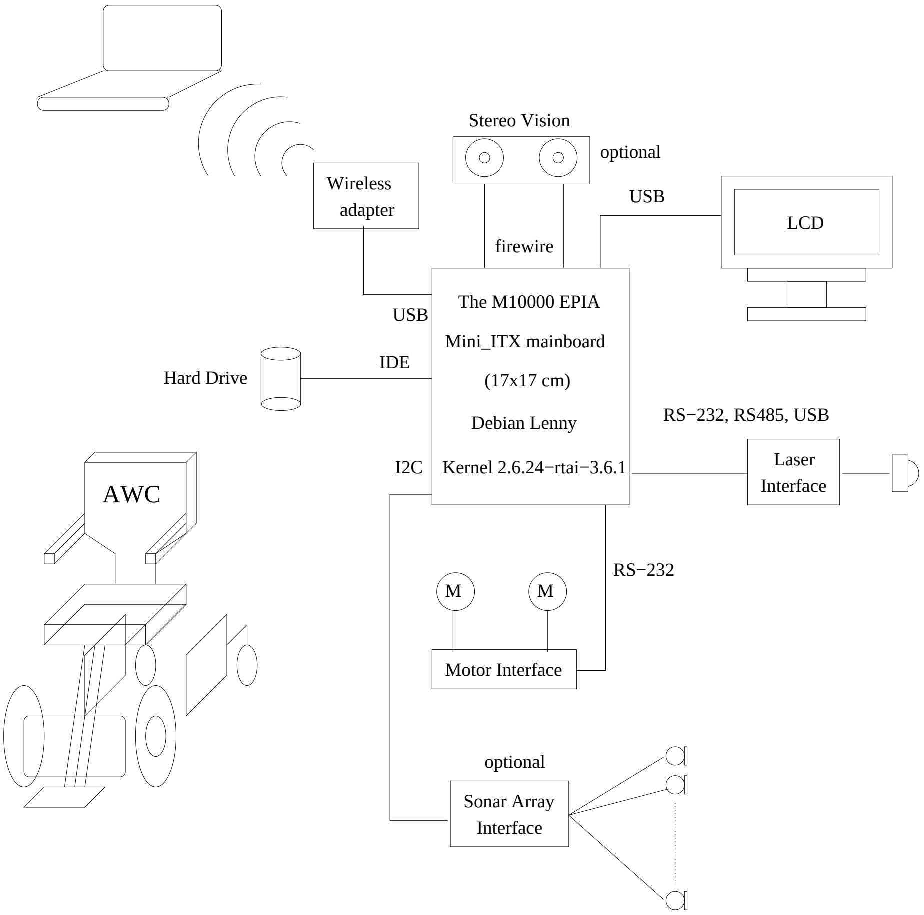


Figure 2.1: The system mainly consists of the following: a) A mini ITX board, in which the kernel 2.6.24-rtai-3.6.1 is running. b) A laptop DELL XPS-M1530 where Debian Lenny stable has been installed and the kernel 2.6.26-2-686 is running The Laptop communicates to the mini board over ssh protocol. c) Two DC motors and a motor interface over RS-232. d) The vision system and the sonar array are optional, and are tehrefor not attached to the system.

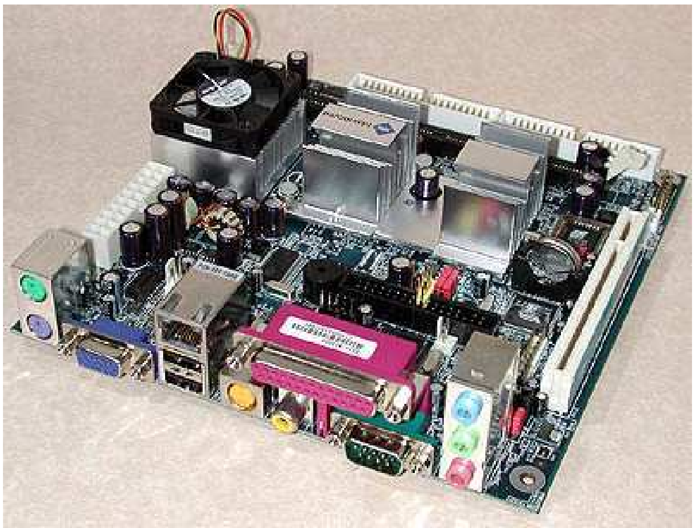


Figure 2.2: EPIA M10000 Mini ITX Mainboard

2.2 DC motors

Lynxmotion GHM-13 Spur Gear Head Motor has been chosen because it is a small and powerful motor and it is perfect for small robotics. With its integrated reduction box, it is specially designed to operate at low speed. This gear motor is ideal for robot propulsion, and it is not necessary to build the reduction system. With a motor in each wheel and a motor controller, the movements of the robot can be controlled.

The specifications of the motor is depicted in table 2.2, and Figure 2.3 shows the motor.

Table 2.2: Lynxmotion DC Motor

parameter	Motor
Voltage	12vdc
RPM	152
Torque	231.05 oz.in (16.7 kg-cm)
Reduction	50:1
Shaft Diameter	6mm
Motor resistance (R)	40 Ω
Friction coeficient (b)	0.014334
Torque constant (K)	0.75890073

2.3 Laser

URG-04LX-UG01 laser range finder was selected because of its size and price. It has a sensing range of 5.6 meters. Measurement accuracy is within 3 percent tolerance of the current reading for most of the sensor's range. And, the scanning

Figure 2.3: Lynxmotion 12 VDC Spur Gear Head Motor.

rate is 100 milliseconds across a 240 degree range. These specifications make the laser ideal for the project in indoor applications.

The specifications is depicted in table 2.3, and the laser together with its scanning range is depicted in figure 2.4.

Table 2.3: URG-04LX-UG01 laser

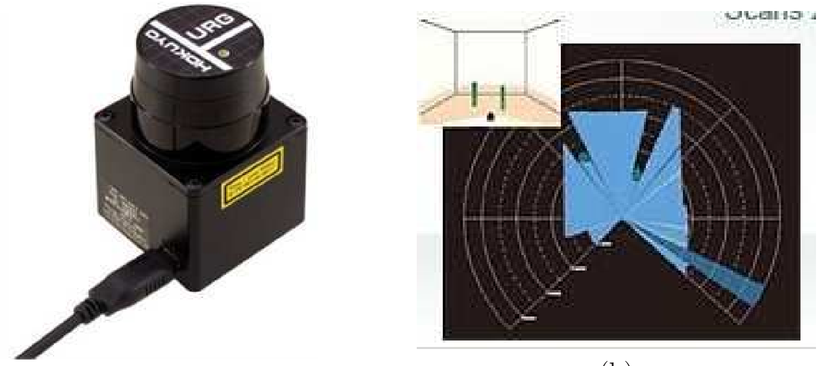
parameter	Laser
Field of view	240
Scanning range	5.6 m
Data interface	RS-232, USB
Supply voltage	5 VDC
Power consumption	500mA at 5V.
Weight	160 g
Dimensions	40×40×70 mm

2.4 Motor's interface

Two motor's interfaces have been build to allow communication to the main board over a serial port. One interface is for reading the velocity from the motors (Motor Input Interface), and the second one is for sending control commands to the motors (Motor Output Interface or Motor controller).

2.4.1 Motor's Input Interface

The block diagram of the motors's input interface is depicted in figure 2.5 The $OPB703$ consists of an infrared emitting diode and a NPN silicon phototransis-



(a) (b)

Figure 2.4: (a) Hokuyo URG-04LX-UG01 laser range finder. (b) Laser response.

tor mounted side by side on a covering optical axis in a black plastic housing. This sensor reads the encoder pattern. The output of the $OPB703$ is amplified by the operational amplifier $LM328$, giving a square signal, which period corresponds to the velocity of the wheel. The $74LS590$ is a 8 bit counter; it counts the pulses, and converts them into a hexadecimal number. The PIC microcontroller reads the hexadecimal that could correspond to the right or left motor, based on an instruction from the board over a serial communication. A 3D representation of the input interface circuit board is shown in figure 2.6.

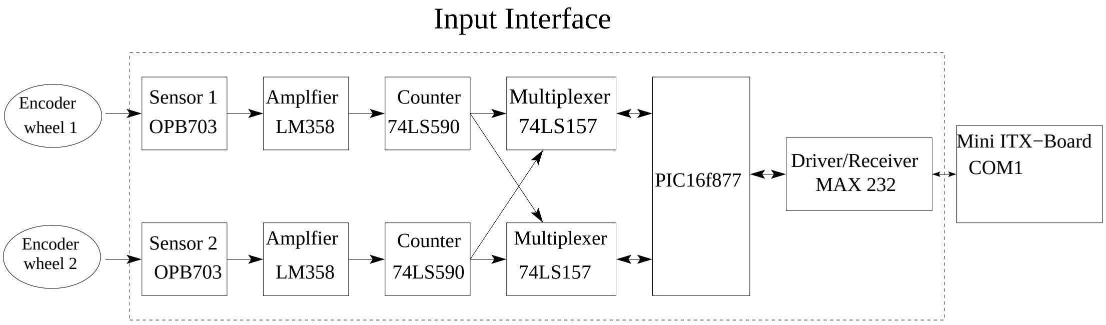


Figure 2.5: Motor Input Interface.

2.4.2 Motor Output Interface

The block diagram of the motor's output interface is depicted in figure 2.7. The board sends commands or instructions to the motors over a serial communication. The PIC microcontroller forwards the instructions from the board to the motors. First, it initializes the $IXDP610$, which is in charge of generating the Pulse Width Modulation (PWM) which period corresponds to the voltage will be applied to the motors. Then, the L293 generates the necessary current to the motors in order not to burn the PIC and the IXDP610s. A 3D representation of the input interface circuit board is shown in figure 2.8.

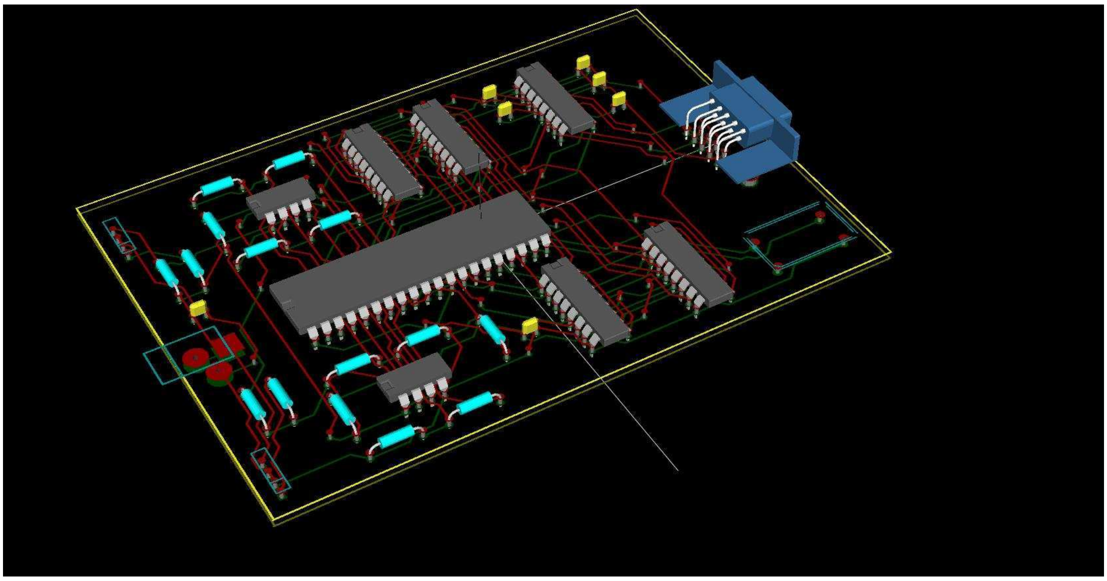


Figure 2.6: 3D representation of the input interface board.

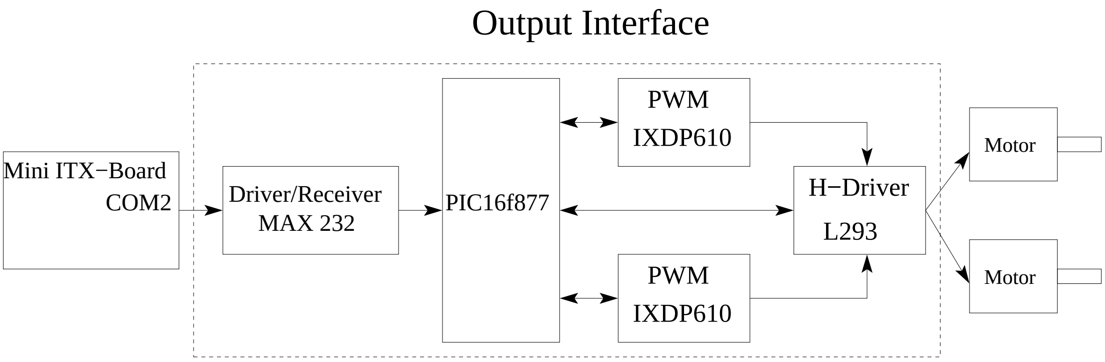


Figure 2.7: Motor Output Interface or motor controller.

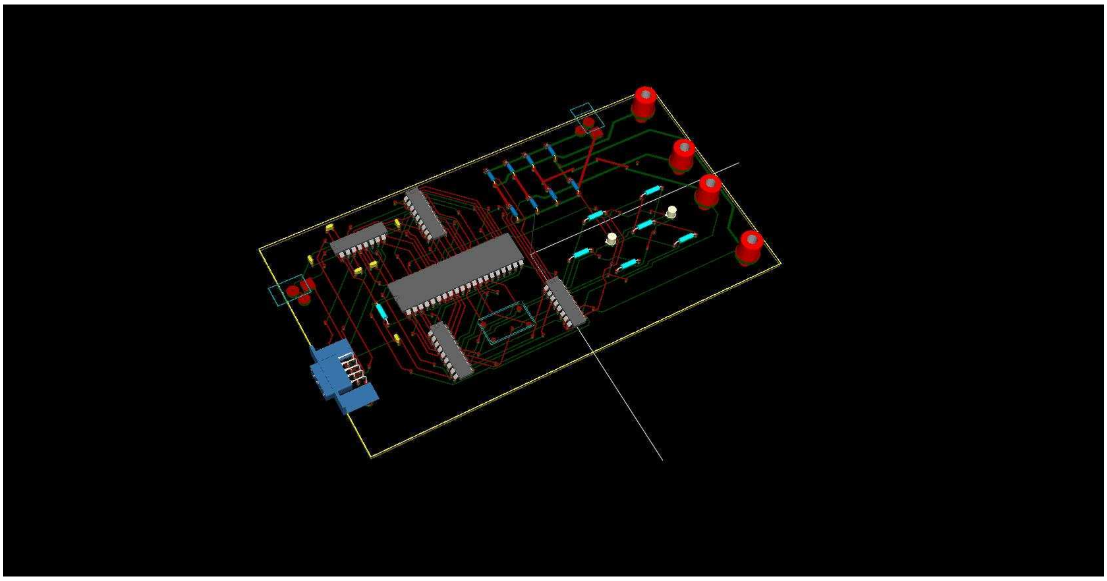


Figure 2.8: 3D representation of the output interface board.

A schematic diagram of the input and output interfaces can be seen in appendix C

Figure 2.9 shows the wheelchair prototype that has been built for testing the high autonomy control level.

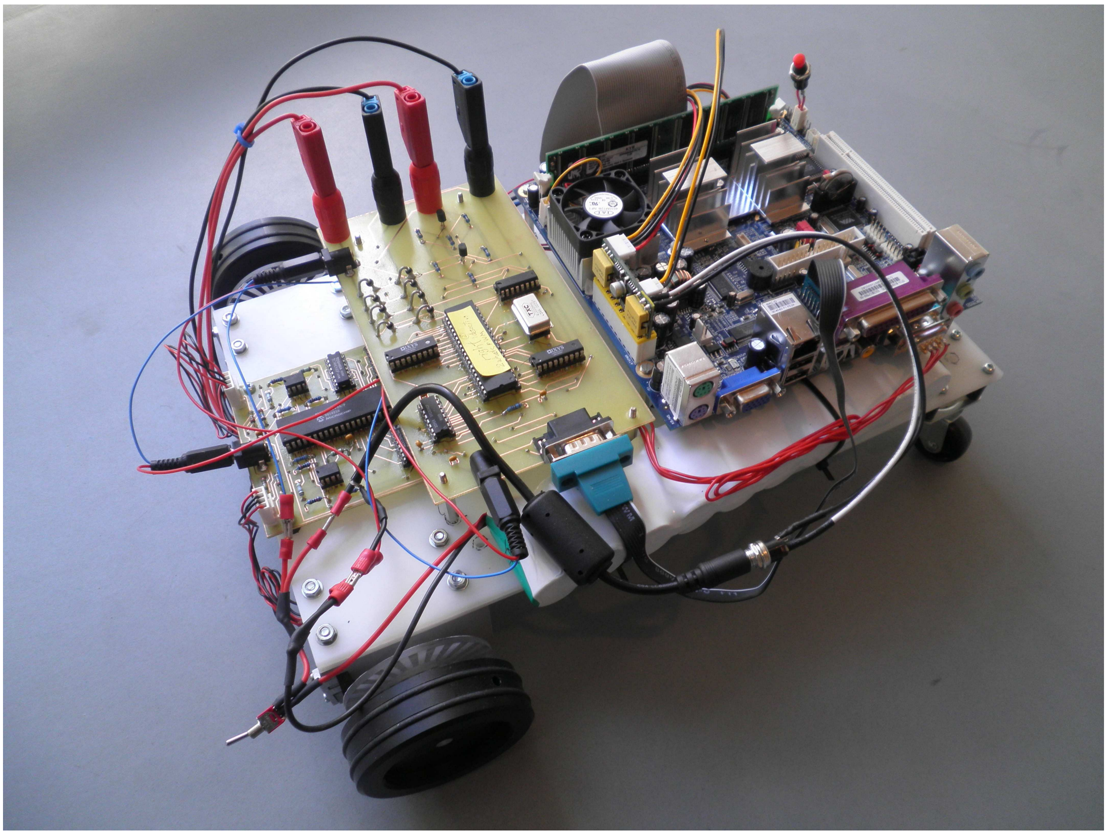


Figure 2.9: Differential mobile robot based wheelchair type.

Part II

Theoretical Background

Chapter 3

Sensor Fusion and Map Making

This chapter outlines the map making and sensor fusion problem based on the Bayesian method. Laser sensor readings are fused using the former method in order to construct a local as well as a global map of the mobile entity, in this case a wheelchair. The method uses occupancy grids to build up a 2D map of the environment. In the following sections the Bayesian inference theorem, the occupancy grids and how both are applied to generate a map, are explained.

3.1 Bayesian Inference

Bayesian inference is a statistical inference method in which observations (evidences) are used to update or infer (conclude) the probability that a hypothesis may be true. The name Bayesian inference comes from the use of Bayes' theorem in the inference of the process. Bayesian inference is an approach to statistics in which all forms of uncertainty are expressed in terms of probability. Bayes' theorem is also known as Bayes' rule [38, 39, 40, 41, 42].

Theorem 3.1.1 (Bayes' rule) *If the events $B_1, B_2, ..., B_k$ constitute a partition in the sample space S, where $P(B_j \neq 0)$ for $j = 1, 2, ..., k$, then for any event A in S such that $P(A) \neq 0$.*

$$P(B_i|A) = \frac{P(B_i \cap A)}{\sum_{j=1}^{k} P(B_j \cap A)} = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{k} P(B_j)P(A|B_j)} \tag{3.1}$$

for $i = 1, 2, ..., k$

Proof 3.1.1 *Appendix A*

- B_i is one of the i mutually exclusive (disjoint) events to be estimated.
- A is the evidence event.
- $P(B_i)$ is the prior probability of the event B_i. It is "prior" in the sense that it does not take into account any information about A.

- $P(B_i|A)$ is the conditional probability of B_i given A. It is also called the posterior probability because it is derived from or depends upon the specified value of A.
- $P(A|B_i)$ is the conditional probability of A given B_i.
- $\sum_{i=1}^{k} P(B_j \cap A) = \sum_{j=1}^{k} P(B_j)P(A|B_j)$ is the total probability of A, and acts as a normalising factor.

3.2 Occupancy Grids

Occupancy grids is a grid-based approach, and it was proposed by [43, 44, 45, 46].

Definition 3.2.1 (Occupancy grids)
An occupancy grids $O_g = \{C_1, ..., C_N\}$ is a tessellation of the robot's environment $\mathcal{W}$ into N cells C_i defined over a discrete spatial lattice.

The cells C_i are stochastic random variables that can take two values, occupied (o) or empty (e).

The state of the cells are exhaustive and exclusive, meaning that $P^o_{i,j} + P^e_{i,j} = 1$, where $P^o_{i,j}$ is the probability of a particular cell on the grid being occupied $C^o_{i,j}$, and $P^e_{i,j}$ is the probability of a particular cell being empty $C^e_{i,j}$. The probability values for the parameters (o, e) are assumed to be independent of each other. Occupancy grids has been applied successfully in robot navigation tasks. The occupancy grids parameters are most often chosen to be probability distributions. The Bayesian fusion algorithm can be used with such probability distributions. A sensor model is needed to represent the uncertainty values of the parameters.

3.3 Application of Bayes Theory to the Occupancy Grid

3.3.1 Recursive Bayes Update Rule

The attraction of the Bayesian inference approach to map building stems from the fact that Bayes' updating rule is recursive. When it is used to support sensor fusion, Bayes' rule provides a way of computing a posteriori probability of a hypothesis being true giving supporting of evidence. [66, 67, 68] have successfully used Bayes' rule to update the occupancy grid for multiple sensor readings $(s_1, ..., s_n)$.

Equations 3.2 and 3.3 are obtained when Bayes' rule from theorem 3.1.1 is transfered to the occupancy grid framework for multiple sensor readings.

$$P^{o|s}_{i,j} = \frac{P^{s|o}_{i,j} P^o_{i,j}}{P^{s|o}_{i,j} P^o_{i,j} + (1 - P^{s|o}_{i,j})(1 - P^o_{i,j})} \tag{3.2}$$

$$P^{e|s}_{i,j} = \frac{P^{s|e}_{i,j} P^e_{i,j}}{P^{s|e}_{i,j} P^e_{i,j} + (1 - P^{s|e}_{i,j})(1 - P^e_{i,j})} \tag{3.3}$$

The following statements are defined.

- The relevant evidence A is given by the sensor reading s.
- The certainty of the true parameter B_i is given by $P^o_{i,j}$ and $P^e_{i,j}$, meaning that they are the prior probabilities of the cell $C_{i,j}$ being occupied or empty. They are taken from the existing map.
- The conditional probability $P(A|B_i)$ is given by $P^{s|o}_{i,j}$ and $P^{s|e}_{i,j}$, which are the conditional probabilities that a sensor reading will exist given the state of the cell $C_{i,j}$, being occupied or empty. This conditional probability is given by the probabilistic sensor model.
- The conditional probability $P(B_i|A)$ is given by $P^{o|s}_{i,j}$ and $P^{e|s}_{i,j}$, which is the conditional probability that a cell is occupied based on the past sensor readings. It is the new estimate.

A new sensor reading s, introduces additional information about the state of the cell $C_{i,j}$. This information is done by the sensor model $P^{s|o}_{i,j}$, and it is combined with the most recent probability estimate stored in the cell. This combination is done by the recursive Bayes' rule $(P^{o|s}_{i,j})$ based on the current set of readings to give a new estimate $P^{o|s}_{i,j}$. It is worth noting that when initialising the map an equal probability to each cell $C_{i,j}$ must be assigned. In other words, the initial map cell prior probabilities are $P^o_{i,j} = P^e_{i,j} = \frac{1}{2}$.

The graphical interpretation of equations 3.2 and 3.3 for a single cell $C_{i,j}$ within a sensor model can be depicted as in Figure 3.1. In order to update the cell $C_{i,j}$ with equations 3.2 and 3.3, a prior probabilities $P^{o_s}_{i,j}$ and $P^{e_s}_{i,j}$ given by the existing grid are needed. These probabilities are combined with the probabilities given by the sensor model ($P^{s|o}_{i,j}$ and $P^{s|e}_{i,j}$) by means of Bayes' rule of combination to give a new estimate or posteriori probabilities ($P^{o|s}_{i,j}$ and $P^{e|s}_{i,j}$).

The following example shows how a single cell state $C_{i,j}$ is updated in the process shown in figure 3.1 through equations 3.2 and 3.3.

Example 3.3.1 *The following terms are given from the existing grid in a particular cell $C_{i,j}$, $P^o_{i,j} = 0.52$, $P^e_{i,j} = 0.48$; update the $P^{o|s}_{i,j}$, $P^{e|s}_{i,j}$ with the terms given by the sensor model $P^{s|o}_{i,j} = 0.56$, and $P^{s|e}_{i,j} = 0.44$.*

$$P^{o|s}_{i,j} = \frac{(0.56)(0.52)}{(0.56)(0.52) + (1 - 0.56)(1 - 0.52)}) = \frac{0.2912}{0.5024} \cong 0.58$$

$$P^{e|s}_{i,j} = \frac{(0.44)(0.48)}{(0.44)(0.48) + (1 - 0.44)(1 - 0.48)}) = \frac{0.2112}{0.5024} \cong 0.42$$

Note that the addition of $P^{o|s}_{i,j}$ and $P^{e|s}_{i,j}$ equal the value of one.

In this report, a total of 30 measurements along a trajectory have been carried out by a mobile robot. In each measurement the laser scans a total of 361 readings distributed along 180^o. Then, each laser reading is interpreted by a probabilistic sensor model as described previously. Afterwards, the recursive Bayes method 3.2 and 3.3 is applied to the data to fuse and update the probabilistic grid map.

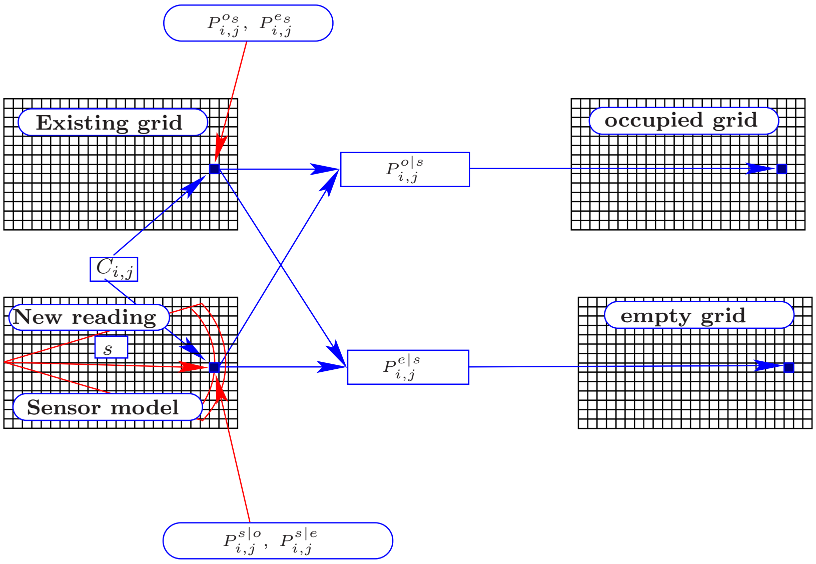


Figure 3.1: A graphical interpretation of equations 3.2 and 3.3 for a single cell $C_{i,j}$ withing the main lobe of the sensor model. An existing grid contains the old probabilities (prior) of the sensor model $P_{i,j}^{o_s}$ and $P_{i,j}^{e_s}$ of a single cell $C_{i,j}$ being occupied or empty respectively. A new sensor data interpreted by a sensor model ($P_{i,j}^{s|o}, P_{i,j}^{s|e}$) is used together with the existing probabilities in the grid to estimate the new state of the cell ($P_{i,j}^{o|s}, P_{i,j}^{e|s}$).

Chapter 4

Localization

4.1 Monte Carlo Localization Algorithm

Over the years, particle filters have been applied with great success to a variety of state estimation problems, [29]. The Monte Carlo Localization (MCL) algorithm is a type of particle filter which is obtained by a proper substitution of the probabilistic motion and perceptual models into the algorithm of particle filter, [33]. A proper review of particle filters and MCL algorithm can be viewed in the following references, [27, 28, 29, 30, 31, 32, 33].

The MCL algorithm can be used to estimate a global localization and tracking position of a wheel mobile entity. However, a drawback in MCL is that once the localization is done it keeps sample sets of a fixed size during time, this fact can make the algorithm inefficient because of waste of computational resources. To overcome the MCL algorithm drawback, it is necessary that once the global localization is achieved by a high number of particles, only a small fraction of particles be used to keep track of the position.

4.2 Kullback Leibler Divergence-sampling

The Kullback Leibler Divergence (KLD) sampling is a variant of of MCL in the way that it adapts the numbers of samples over time. This means that if the state estimate is accurate enough, the number of particles are reduced. On the other hand, if the state estimate is not good enough, the number of particles are increased. A mathematical derivation is not proved here. For that, take a look at [33]. An explanation of the algorithm is stated instead, and some simulations are shown in chapter 11.

In order to achieve the localization in this study, 18 of 361 laser readings are chosen in each measurement. They are equally distributed along the half front plane of the robot, and afterwards taken by the KLD to perform the localization for each robot pose during the trajectory.

The method is shown in algorithm 4.1. It takes as inputs the previous sample set X_{t-1}, the map, the most recent control odometry data U_t and laser measurements Z_t. It also takes the statistical error bounds ϵ and δ. In contrast, it ensures the actual sample set X_t and the best state estimate x_{est}.

Lines 1 through 2 initialize X_t, M and M_x. Initially each bin b in the histogram H is set to empty, lines 3 through 5. In line 7, a particle is drawn from the previous particle set. In other words, the particle is drawn in a probabilistic way according to the weights of the particles w^i_{t-1}. The outcome is a single particle. This particle is then predicted, weighted and inserted in the new sample set (lines 8-10). The core of the KLD-sampling is implemented in lines 11 through 18. Line 11 examines whether the new particle falls into an empty bin in the histogram. If it does, then the number k of non-empty bins is incremented and then the bin is set to non-empty bin. So, the number k has to do with the number of histogram bins filled with at least one particle. M_x gives the number of particles needed based on the number k and the statistical error bounds ϵ and δ, these values are available in standard statistical tables. The algorithm main loop ends up when M exceeds M_x and M_{min}. Lines 20 through 23 normalize the weights to ensure a probabilistic distribution. Lines 23 through 24 selects the best state estimate x_{est}.

A reasoning of formula in line 15 is as follows: In the early stages of sampling, k increases with almost every new sample since every bin in the histogram is practically empty. This fact makes M_x increase. However, over time less and less bins are empty making the increasing rate of M_x slow down. Since M increases with every new sample, there will be a point in time where M will exceed M_x and M and the sampling stops. During tracking the algorithm generates less samples since the particles are concentrated in a small numbers of different bins, [33].

Algorithm 4.1 KDL-Sampling

```
Require: $X_{t-1},\ U_t,\ Z_t,\ map,\ \epsilon,\ \delta$
Ensure: $X_t,\ x_{est}$
$X_t = 0$
$M = 0, M_x = 0, k = 0$
for all $b$ in $H$ do
    $b$ =empty
end for
while $M < M_x\ AND\ M < M_{min}$ do
    draw a particle $i$ with a probability $w_{t-1}^{[i]}$
    $X_t^{[M]} =$ sample_motion_model$(U_t, x_{t-1}^i)$
    $w_t^{[M]} =$ measurement_model$(z_t, x_t^M, map)$
    $X_t = X_t + \langle X_t^M, w_t^M \rangle$
    if $x_t^M$ falls into an empty bin $b$ then
        $k = k + 1$
        $b$ =non-empty bin
        if $k > 1$ then
            $M_x := \frac{k-1}{2\epsilon}\left\{1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}} z_1 - \delta\right\}^3$
        end if
    end if
    $M = M + 1$
end while
$W_{total} = \sum_{i=1}^{M} w_t^{[i]}$
for $i = 1$ to $M$ do
    $w_t^{[i]} = \frac{w_t^{[i]}}{W_{total}}$
end for
$index = max(W_{total})$
$x_{est} = X_t(index)$
```

Chapter 5

Motion Planning

5.1 Basic Problem and its Simplification

Simplification of the basic motion planning problem is treated in [17] and [25]. The notions are presented in this section. The idea of the basic motion planning problem is to isolate some central issues before considering some additional difficulties, i.e. before it can be expanded to more complex and realistic scenarios. In the basic problem, the robot is the only moving object in the workspace and the dynamic properties of the robot are ignored. Motion is restricted to non-contact motion, so mechanical interaction between robot and obstacles can be ignored. The geometric form of the robot as well as the kinematic restrictions are not consider either. These assumptions transform the "physical motion planning" problem into a purely geometrical path planning problem [17].

The basic motion planning problem resulting from the mentioned simplifications can be stated as follows:

- Let $\mathcal{A}$ be a single rigid object (the mobile robot), which is often modelled as a point.
- The environment of the mobile robot is static and known.
- The robot is moving in an Euclidean space $\mathbb{R}^N$, with $N = 2$ or 3, the Euclidean space also called *workspace* $\mathcal{W}$
- Let $\mathcal{B}_1, ..., \mathcal{B}_q$ fix rigid objects in $\mathcal{W}$. The $\mathcal{B}_i's$ are called obstacles and their location in $\mathcal{W}$ are accurately known.
- No kinematic constraints limits the motion of $\mathcal{A}$ ($\mathcal{A}$ is a free flight object).
- A path τ is generated from an initial position and orientation to a goal position and orientation. The path τ specifies a continuous sequences of positions and orientations of $\mathcal{A}$ in $\mathcal{W}$ avoiding contact with the $\mathcal{B}_i's$.

5.2 Configuration Space of the Robot

Let the Mobile Robot $\mathcal{A}$ be described by the vector $q = [x, y, \theta]$ which is a compact set (closed and bounded) in $\mathcal{W}$. Let $\mathcal{W}$ be in $\mathbb{R}^2$ and the obstacles

$\mathcal{B}_1, ..., \mathcal{B}_q$ be closed subsets of $\mathcal{W}$. $\mathcal{F}_\mathcal{A}$ and $\mathcal{F}_\mathcal{W}$ are Cartesian frames; $\mathcal{F}_\mathcal{A}$ is fixed in $\mathcal{A}$, but movable in $\mathcal{F}_\mathcal{W}$. $\mathcal{F}_\mathcal{W}$ is embedded in $\mathcal{W}$. The obstacles $\mathcal{B}_i$ are fixed with respect to $\mathcal{F}_\mathcal{W}$. The vector q represents the configuration of $\mathcal{A}$ with respect to $\mathcal{F}_\mathcal{W}$, where (x, y) represents the position and θ the orientation.

Definition 5.2.1 (configuration space)*[17]*
A configuration of $\mathcal{A}$ is the space $\mathcal{C}$ of all possible configurations of $\mathcal{A}$.

Remark 5.2.1 *$\mathcal{A}(q)$ is a subset of $\mathcal{W}$ occupied by $\mathcal{A}$ at the configuration q. All the possible configurations of $\mathcal{A}$ forms the configuration space $\mathcal{C}$, e.q. it is a set of all possible values of $q = [x, y, \theta]$.*

In other words, the configuration space or $\mathcal{C}$-space of the mobile system, in this case a wheelchair, is the space of all possible configurations of the system. Thus, a configuration is simply a point in this abstract configuration space, [21].

5.3 Obstacles in Configuration Space

The obstacles $\mathcal{B}_1, ..., \mathcal{B}_q$ must be considered in the configuration space of the robot, which means that they must be mapped onto this configuration space. Feasible paths must take into account the position of the obstacles in the configuration space of the robot. Therefore, a subset $\mathcal{C}$ is described which is made of contact-free configurations. In other words, when the robot is moving in $\mathcal{W}$ and if it finds any obstacle $\mathcal{B}_i$ which has been configured on $\mathcal{W}$, the robot $\mathcal{A}$ must avoid collision with them, i.e. the robot cannot move into an obstacle.

Definition 5.3.1 (C-obstacle)*[17]*
Any obstacle $\mathcal{B}_i$, with $i = 1, .., q$, in the workspace $\mathcal{W}$ maps in the configuration space $\mathcal{C}$ to a region $\mathcal{CB}_i$ is called $C - obstacle$, so that

$$\mathcal{CB}_i = \{q \in \mathcal{C} \setminus \mathcal{A}(q) \cap \mathcal{B}_i \neq \emptyset\} \tag{5.1}$$

Equation 5.1 says that the intersection between a particular configuration of the robot $\mathcal{A}(q)$ with an obstacle $\mathcal{B}_i$, with $i = 1, .., q$, is different than the empty set, i.e. there is a collision of the robot with the obstacle.

Definition 5.3.2 (C-obstacle-region)*[17]*
The union of all $C - obstacle$ gives the $C - obstacle - region$ ($\mathcal{CB}_{region}$), so that

$$\mathcal{CB}_{region} = \bigcup_{i=1}^{q} \mathcal{CB}_i \tag{5.2}$$

Definition 5.3.3 (free space)*[17]*
A free configuration space $\mathcal{C}_{free}$ is a subset $\mathcal{C}$ defined by,

$$\mathcal{C}_{free} = \mathcal{C} | \bigcup_{i=1}^{k} \mathcal{CB}_i = \left\{ q \in \mathcal{C} \setminus \mathcal{A}(q) \cap \left(\bigcup_{i=1}^{q} \mathcal{B}_i \right) = \emptyset \right\} \tag{5.3}$$

Equation 5.3 says that the intersection between a particular configuration space of the robot $\mathcal{A}(q)$ with $\mathcal{CB}_{region}$, is equal to the free space, i.e. there is no collision between the robot and the obstacle.

5.4 Path Planning

Some work in robotics has focused on the use of harmonic functions for robot path planning. Harmonic functions are solution to Laplace's equation, [12, 13, 14, 15, 16]. In other words, path planning is achieve by solving the Laplace's equation to harmonic functions.

Definition 5.4.1 (path)*[17]*
A path τ of the robot $\mathcal{A}$ in the configuration space of the robot $\mathcal{C}$, is a succession of configurations of $\vec{q}$, they go from $\vec{q}_{init}$ (initial configuration) to $\vec{q}_{goal}$ (final configuration). The path must be continuous without a gap in the workspace. In other words, a path is a continuous point map $\tau : [0,1] \rightarrow \mathcal{C}$ connecting $\vec{q}_{init} = \tau(0)$ to $\vec{q}_{goal} = \tau(1)$.

Definition 5.4.2 (Laplace's equation)*[13]*
A harmonic function ϕ on a domain $\Omega \subset R^n$ is a function that satisfies Laplace's equation

$$\Delta^2 \phi = \sum_{i=1}^{n} \frac{\partial^2 \phi}{\partial x_i^2} = 0 \tag{5.4}$$

[12] uses the Finite Difference Method (FDM) for the solution of Laplace's equation and [15] uses Taylor series to solve 5.8. In both cases the solution yield to the equation 5.5

$$\phi_{i,j} = \frac{1}{4}(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}) \tag{5.5}$$

Corollary 5.4.1 *[12]*
If ϕ satisfies Laplace's equation, then at any point in the domain Ω, is the average of the values at four surrounding points.

For planning purposes, the grid elements that represent boundary conditions like the obstacles and goal are held fixed. This method is recursive, in the way that, as soon as a new average interaction is available, it is placed in the former average interaction place. The interaction process ends up when there is no change of any cell grid from one interaction to the next. The algorithm does not suffer from local minima as the potential field algorithm does, [17, 18, 16, 19, 20, 21, 22]. One important feature of harmonic functions is that they can be used to generate smooth, collision-free paths without the threat of spurious of local minima, [24]. The process can be seen in figure 5.1. And the algorithm is shown in 5.1.

5.5 Circular Entity

A mobile entity is considered a circular shape, that limits the configuration space to $\mathbb{R}^2$, [21]. Motion planning is now equivalent to motion planning for a point in the configuration space. Figure 5.2 shows how a workspace obstacles is transformed to a configuration space obstacles.

The morphological operation called dilation [26] can be used to implement the configuration space.

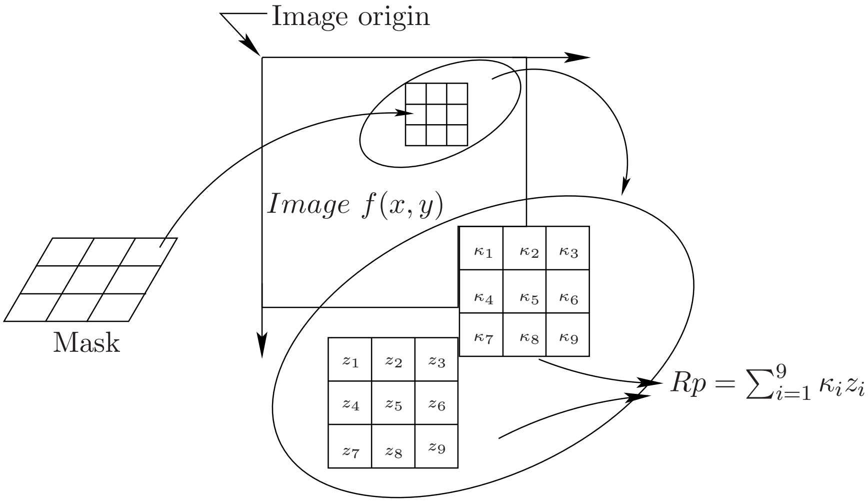


Image origin
Image $f(x,y)$
Mask
κ_1 κ_2 κ_3
κ_4 κ_5 κ_6
κ_7 κ_8 κ_9
z_1 z_2 z_3
z_4 z_5 z_6
z_7 z_8 z_9
$Rp = \sum_{i=1}^{9} \kappa_i z_i$

Figure 5.1: The showed cell in the grid (x, y) is replaced by the new interaction $\phi_{i,j}$. This process is repeated for all cells in the grid.

Algorithm 5.1 Path planning using Laplace's equation

```
**Require:** $\phi^0_{i,j}$ {The initial map };
**Ensure:** $\phi^{n+1}_{i,j}$ {Harmonic functions};
  $\phi_{i,j} \Leftarrow 1$ {Obstacles $(i, j)$ are fixed to 1};
  $\phi_{i,j} \Leftarrow 0$ {Goal position $(i, j)$ is fixed to 0};
  **while** $\phi^{n+1}_{i,j} \neq \phi^n_{i,j}$ **do**
    $\phi^{n+1}_{i,j} \Leftarrow \frac{1}{4}(\phi^n_{i+1,j} + \phi^n_{i-1,j} + \phi^n_{i,j+1} + \phi^n_{i,j-1})$
  **end while**
```

Definition 5.5.1 (dilation)*[26]*
With A and B be sets in Z^2, where Z denotes the set of real integers, the dilation of A and B denoted $A \oplus B$ is defined as:

$$A \oplus B = \{z | (\hat{B})_z \cup A \neq \varnothing\} \tag{5.6}$$

The dilation depends of the translation and the reflection which are defined in the following:

Definition 5.5.2 (translation)*[26]*
The translation of a set B by a point $z = (z_1, z_2)$, denoted $(B)_z$, is defined as

$$(B)_z = \{c | c = a + z, \ for \ a \ \in B\} \tag{5.7}$$

Definition 5.5.3 (reflection)*[26]*
The reflection of a set B, denoted $\hat{B}$, is defined as

$$\hat{B} = \{w | w = -b, \ for \ b \ \in B\} \tag{5.8}$$

In other words, the dilation is obtained by the reflection of B about its origin and then shifted by z, A and B must overlap at least one element. Figure 5.3 shows the dilation of a set B by a structuring element B which is symmetric.

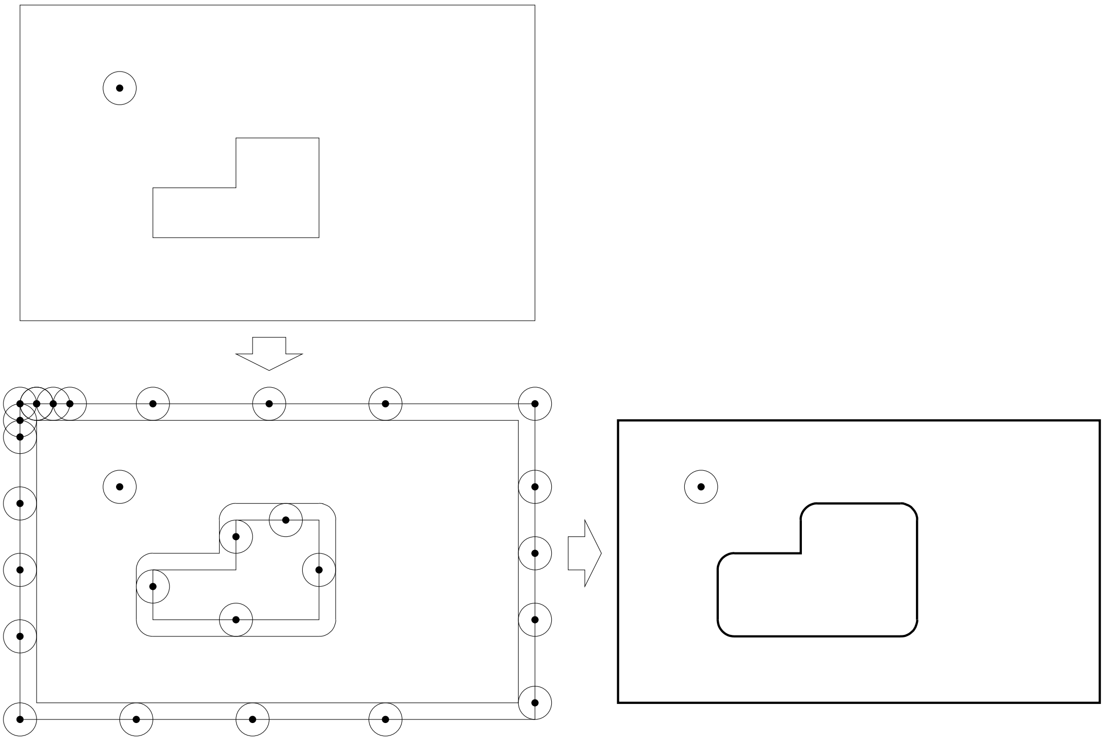


Figure 5.2: The top left figure shows the work space obstacle and the circular mobile entity. Left bottom slides the mobile entity around the obstacle and keeps track of the curve traced out by the reference point. Bottom right shows the configuration space. Motion planning in the work space has been transformed to motion planning in the configuration space.

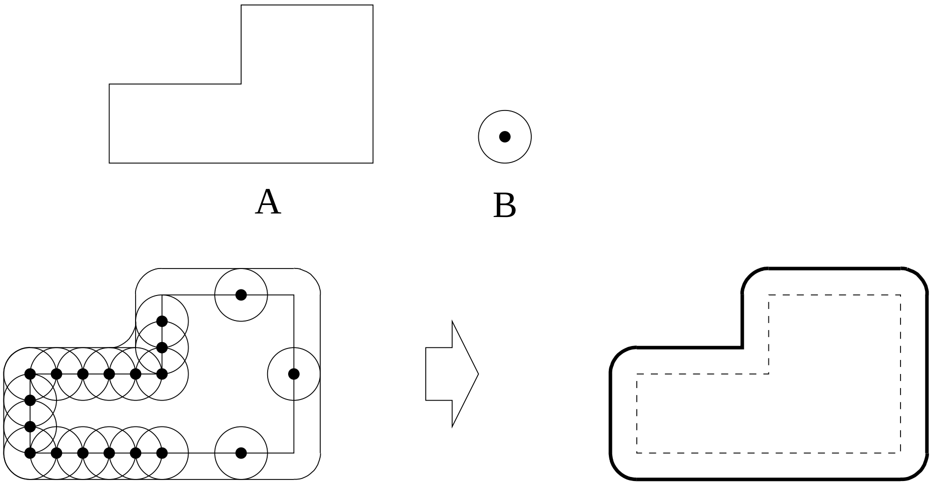


Figure 5.3: Shows the set A and the structuring element B. The structuring element and its reflection are the same because B is symmetric with respect to its origin. The dash line is the original set and the solid line is the dilation of the set A by B.

Chapter 6

Mathematical Model of a Wheelchair

This chapter deals with the computation of a kinematic and dynamic model of a differential wheelchair. The main objective in the development of the kinematic model is to find a mapping matrix that relates velocities in the mobile entity to velocities in a cartesian coordinate system.

To this end, a kinetic energy of a wheel with respect to its center of gravity and also a kinetic energy of the main body of the system are computed separately and then added to obtain the total kinetic energy of the system.

[53, 54, 55, 56, 57]

6.1 Geometry of the Differential Drive Wheelchair

In order to illustrate the methodology, the differential drive wheelchair geometric model, as shown in figure 6.1, is considered. In this model the following is defined.

- $\mathcal{F}_{\mathcal{A}}$ is the fixed robot's reference frame with coordinates variables (x_1, x_2).
- $\mathcal{F}_{\mathcal{W}}$ is the world reference frame with coordinates variables (x, y).
- CM is the center of mass of the frame.
- P_c is the point of the center of mass with coordinates (x_c, y_c)
- P_l is a virtual reference point with coordinates (x_l, yl).
- P is an intersection point between the driving wheel axis and the geometry axis of symmetry.
- b is the distance between the center of each wheel and the geometry axis of symmetry.
- a is the length of the platform and in the direction perpendicular to the driving wheel axis.
- d is the distance from P_c to P in the direction of the axis of symmetry.

- L is the distance from P to P_l in the direction of the axis of symmetry.
- r is the radius of each wheel.
- w_r is the right wheel.
- w_l is the left wheel.
- θ is the angle of rotation.

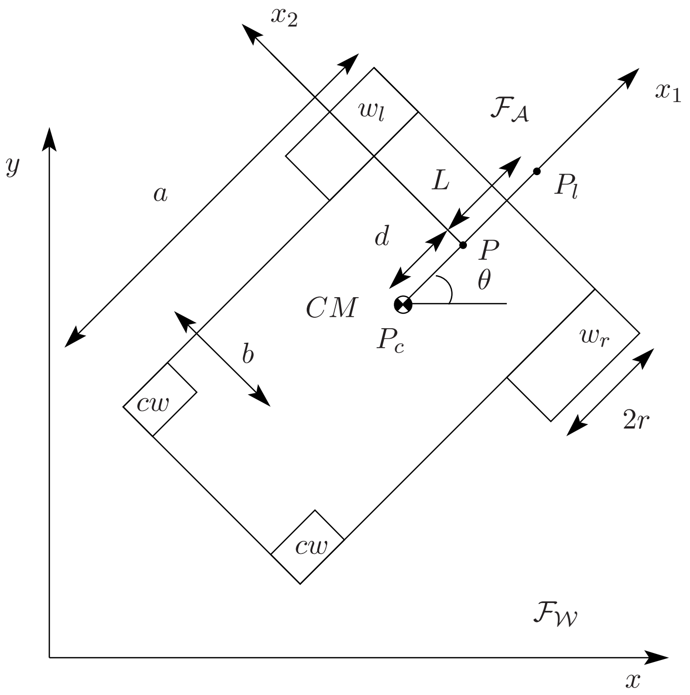


Figure 6.1: Differential drive wheelchair geometry. A fixed local frame $\mathcal{F}_{\mathcal{A}}$ is attached to the mobile robot and moving with respect to a world reference frame $\mathcal{F}_{\mathcal{W}}$. θ is the angle of rotation around the center of mass CM and with respect to $\mathcal{F}_{\mathcal{W}}$. The robot moves along x_1 and perpendicular to the driving wheel axis.

6.2 Constraint Matrix

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

Definition 6.2.1 (Constraint Matrix)*[50]*
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,

$$A(q)\dot{q} = 0 \tag{6.1}$$

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

In definition 6.2.1, $A(q)$ is the constraint matrix and some remarks can be stated.

Remark 6.2.1 • *It is a holonomic constraint, if either a constraint equation is of the form $C(q)$ or it can integrated.*

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

6.3 Constrains

Two types of constrains are imposed to the fixed wheels of the system

6.3.1 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, [49]. In other words, the forward velocity of the center of mass with respect to a fixed world frame, $\mathcal{F}_{\mathcal{W}}$, must be equal to the linear velocity of the wheel. This situation can be depicted in figure 6.2, and can be expressed mathematically as stated in equation 6.2

$$\dot{x}_c cos\theta + \dot{y}_c sin\theta + b\dot{\theta} = r\dot{\psi} \tag{6.2}$$

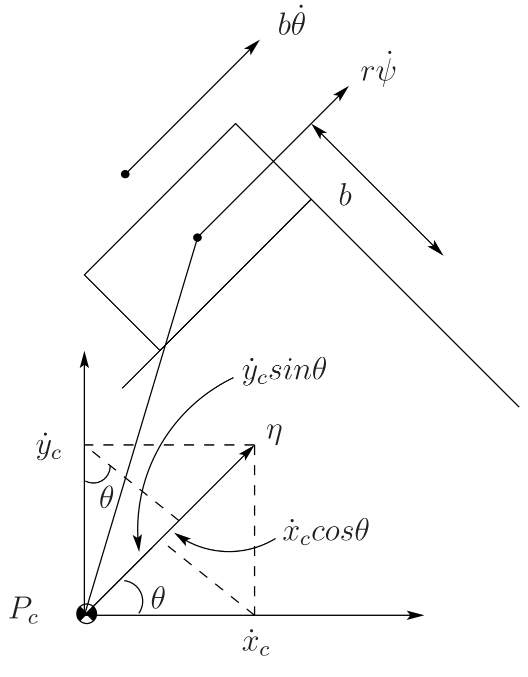


Figure 6.2: Shows the rolling constraint without slipping

6.3.2 No lateral Movement

This constrain assumes that the wheel's orthogonal components are zero, [49]. This situation is depicted in figure 6.3 and represented mathematically in equation 6.3.

$$\dot{y}_c cos\theta - \dot{x}_c sin\theta - d\dot{\theta} = 0 \tag{6.3}$$

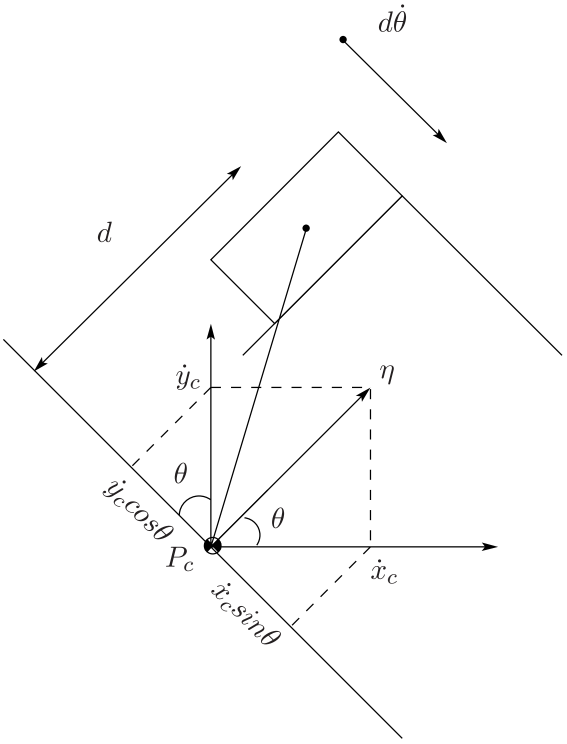


Figure 6.3: Shows the lateral constraint

6.4 Kinematic Model of a Differential Wheelchair

In this section a formal definition of kinematic is presented. Furthermore, a kinematic model of a differential wheelchair is shown.

Definition 6.4.1 (Kinematics)*[47]*
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 6.4.2 (Kinematic Model)*[48]*
From the mechanical system given by equation 6.1, Let $S_i = [s_1, \cdots, s_{n-m}]^T$ be a

set of smooth (continuously differentiable) and linearly independent vector fields in the null space of $A(q)$, $(N(A))$, such that, $A(q)S_i(q) = 0, \quad i = 1, \cdots, n-m$. Hence, $S_i(q) = \dot{q}$. Now it is possible to define $(n-m)$ velocities $\eta(t) = [\eta_1, \cdots, \eta_{n-m}]^T$ such that for all t.

$$\dot{q} = S(q)\eta(t) \tag{6.4}$$

Equation 6.4 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 differential drive wheelchair, firstly, a constraint matrix of the form of equation 6.1 must be obtained based on the constraints from equations 6.2 and 6.3. Equations 6.5 to 6.7 are the constraints imposed to the fixed wheels of the system. Equations 6.8 and 6.9 are the constraint matrix and the generalized coordinate vector respectively.

$$\dot{y}_c cos\theta - \dot{x}_c sin\theta + \dot{\theta} d = 0 \tag{6.5}$$

$$\dot{x}_c cos\theta + \dot{y}_c sin\theta + b\dot{\theta} - r\dot{\psi}_r = 0 \tag{6.6}$$

$$\dot{x}_c cos\theta + \dot{y}_c sin\theta - b\dot{\theta} - r\dot{\psi}_l = 0 \tag{6.7}$$

$$A(q) = \begin{bmatrix} -sin\theta & cos\theta & d & 0 & 0 \\ -cos\theta & -sin\theta & -b & r & 0 \\ -cos\theta & -sin\theta & b & 0 & r \end{bmatrix} \tag{6.8}$$

$$\dot{q} = \begin{bmatrix} \dot{x}_c & \dot{y}_c & \dot{\theta} & \dot{\psi}_r & \dot{\psi}_l \end{bmatrix}^T \tag{6.9}$$

Secondly, a system model given by 6.4 that corresponds to a differential wheelchair entity must be derived. To this end, a Jacobian matrix $S(q)$, that satisfies the relation $A(q)S(q) = 0$ must be computed and finally a velocity vector η is stated. The computation of $N(A)$ turns out in equation 6.10, whereas, the velocity vector results in $\eta = [\eta_1, \eta_2]^T = [\dot{\psi}_1, \dot{\psi}_2]^T$.

$$S(q) = [s_1(q), s_2(q)] = \begin{bmatrix} c(bcos\theta + dsin\theta) & c(bcos\theta - dsin\theta) \\ c(bsin\theta - dcos\theta) & c(bsin\theta + dcos\theta) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.10}$$

6.5 Dynamic Model of a Differential Wheelchair

The Lagrangian formalism to holonomic and nonholonomic systems can be found in different sources in the literature [51, 47, 52] and stated in equation 6.11.

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial L(q,\dot{q})}{\partial q} &= M_I(q)\ddot{q} + V(q,\dot{q}) \\ &= B(q)\tau - A^T(q)\lambda \end{aligned} \tag{6.11}$$

Where:

$L(q, \dot{q}) = T(q, \dot{q}) - W(q)$ is the Lagrangian as is 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.

$A^T(q)$ is the Jacobian transport 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.

It is assumed that the differential wheelchair is moving only in a horizontal plane. Therefore, the potential energy remains constant and 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 6.11, 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 two wheels.

6.5.1 Kinetic Energy of the Wheel

Figure 6.4 shows the schematic representation of a single wheel attached to the trolley. In this representation it is assumed, 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 $\mathcal{F}_{\mathcal{W}}$ is found, and then multiplied by the angular density ρ_α. The former can be established formally in the following theorem.

Theorem 6.5.1 (Wheel'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:

$$T_w = \frac{1}{2} \int_0^{2\pi} (\dot{x}_{P_w}^2 + \dot{y}_{P_w}^2 + \dot{z}_{P_w}^2) \rho_\alpha \, \mathrm{d}\psi \tag{6.12}$$

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.

$$P_w = \begin{bmatrix} x_c + lcos(\alpha + \theta) + rcos\psi cos\theta \\ y_c + lsin(\alpha + \theta) + rcos\psi sin\theta \\ rsin\psi \end{bmatrix} \tag{6.13}$$

Taking the square derivative of $\dot{P_w}^2$ and substituting the result in 6.12 and then computing the integration, the kinetic energy of a wheel with respect to the center of mass is obtained.

$$
\begin{aligned}
T_w = {} & \frac{1}{2}m_w\dot{x}_c + \frac{1}{2}m_w\dot{y}_c - m_w l\dot{x}_c\dot{\theta}sin(\alpha+\theta) + m_w l\dot{y}_c\dot{\theta}cos(\alpha+\theta) \\
& + \frac{1}{2}m_w l^2\dot{\theta}^2 + \frac{1}{4}I_w\dot{\theta}^2 + \frac{1}{2}I_w\dot{\psi}^2 \qquad (6.14)
\end{aligned}
$$

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 $m_w = 2\pi\rho_\alpha$.

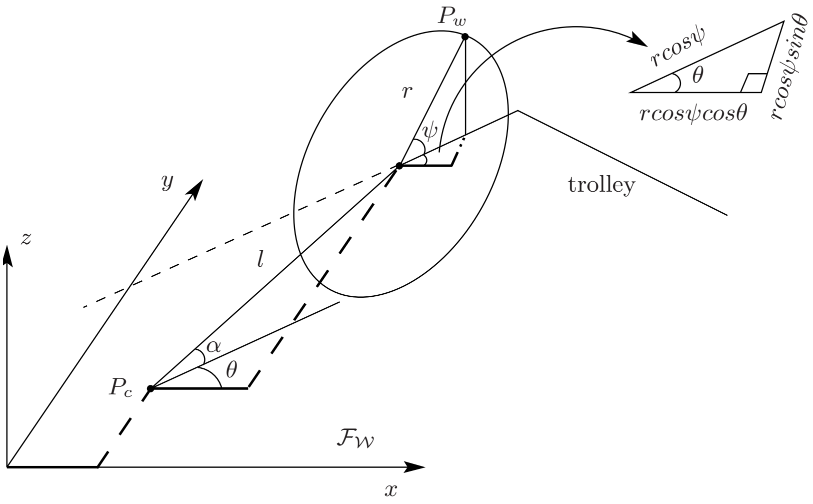


Figure 6.4: A wheel attached to the trolley. The mass m_w is assumed to be distributed evenly with radius r when computing the kinetic energy of the wheel with respect to the center of mass.

6.5.2 Kinetic Energy of the wheelchair's Frame

By combining rotational and translational energies, a kinetic energy of the frame can be obtained. Figure 6.5 shows the wheelchair's frame rotational and translational velocities. The frame's kinetic energy can be stated formally as in the following definition.

Definition 6.5.1 (Frame's kinetic energy)*[47]*
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.

$$T_T = \frac{1}{2}M_T\eta_c^2 + \frac{1}{2}I_T\omega_c^2 \qquad (6.15)$$

Where:
M_T is the mass of the trolley or frame in $[Kg]$.
η_c is the linear velocity of the trolley in $[m/s]$ with respect of its CM.
I_T is the moment of inertia of the frame in $[Kg \cdot m^2]$ with respect of its CM
ω_c is the angular velocity in $[Rad/s]$ of the frame about its CM

Equation 6.15 represents the kinetic energy of a rotational and moving wheelchair frame with respect to its center of mass CM.

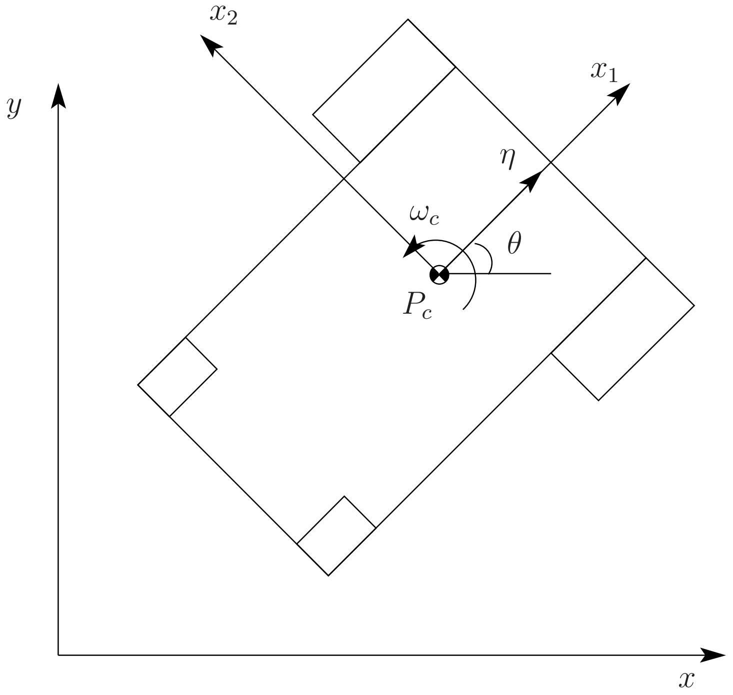


Figure 6.5: Wheelchair that rotates and translates about its center of mass.

6.5.3 Total Kinetic Energy of the wheelchair's Frame

The total kinetic energy of the wheelchair T can be obtained by adding the kinetic energy of the frame 6.15 plus the kinetic energy of the two wheels 6.14

$$T = T_T + \sum_{i=1}^{2} T_{w_i} \tag{6.16}$$

Expanding equation 6.16 gives equation 6.21.

$$\begin{aligned} T = &\frac{1}{2}\dot{x}_c^2\Big(M_T + 2m\Big) + \frac{1}{2}\dot{y}_c^2\Big(M_T + 2m\Big) + \frac{1}{2}\dot{\theta}^2\Big(I_T + 2ml^2 + I_w\Big) \\ &-\dot{x}_c\dot{\theta}\Big[mlsin(\alpha_1 + \theta) + mlsin(\alpha_2 + \theta)\Big] \\ &+\dot{y}_c\dot{\theta}\Big[mlcos(\alpha_1 + \theta) + mlcos(\alpha_2 + \theta)\Big] \\ &+\frac{1}{2}I_w\dot{\psi_1}^2 + \frac{1}{2}I_w\dot{\psi_2}^2 \end{aligned} \tag{6.17}$$

Equation 6.21 is the total kinetic energy of a differential wheelchair with respect to its CM. This equation can be reduced to a matrix notation as stated in equation 6.18

$$T = \frac{1}{2}\dot{\xi}^T R^T M R\dot{\xi} + \frac{1}{2}\dot{\psi}^T I_{wm}\dot{\psi} \tag{6.18}$$

Where:

$$\dot{\xi} = \begin{bmatrix}\dot{x}_c & \dot{y}_c & \dot{\theta}_c\end{bmatrix}^T \tag{6.19}$$

$$R = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} \frac{1}{2}\Big(M_T + 2m\Big) & 0 & -ml\sum_{i=1}^{2} sin\big(\alpha_i + \theta\big) \\ 0 & \frac{1}{2}\Big(M_T + 2m\Big) & ml\sum_{i=1}^{2} cos\big(\alpha_i + \theta\big) \\ 0 & 0 & \frac{1}{2}\Big(I_T + 2ml^2 + I_w\Big) \end{bmatrix}$$

$$I_{wm} = \begin{bmatrix} I_w & 0 \\ 0 & I_w \end{bmatrix}$$

$$\dot{\psi} = \begin{bmatrix}\dot{\psi}_1 & \dot{\psi}_2\end{bmatrix}^T \tag{6.20}$$

6.5.4 Dynamic Equations

In order to derive the dynamic equations of motion of the mobile platform, the total kinetic energy of the system, equation 6.21, is replaced in the left expression of equation 6.11. The Lagrange equations of motion of the differential wheelchair with Lagrangian multipliers λ_1, λ_2 and λ_3, are given by equations 7.1 to 6.25.

$$\big(M_T + 2m_w\big)\ddot{x}_c - 2m_w d\big(\dot{\theta}^2 cos\theta - \ddot{\theta} sin\theta\big) - \lambda_1 sin\theta - \big(\lambda_2 + \lambda_3\big)cos\theta = 0 \tag{6.21}$$

$$\big(M_T + 2m_w\big)\ddot{y}_c - 2m_w d\big(\ddot{\theta} cos\theta - \dot{\theta}^2 sin\theta\big) + \lambda_1 cos\theta - \big(\lambda_2 + \lambda_3\big)sin\theta = 0 \tag{6.22}$$

$$\big(I_T + 2m_w l^2 + I_w\big)\ddot{\theta} + 2m_w d\big(\ddot{y}_c cos\theta - \ddot{x}_c sin\theta\big) + \lambda_1 d + \big(\lambda_2 - \lambda_3\big)b = 0 \tag{6.23}$$

$$I_w\ddot{\psi}_r + \lambda_2 r = \tau_r \tag{6.24}$$

$$I_w\ddot{\psi}_l + \lambda_3 r = \tau_l \tag{6.25}$$

Where, τ_r and τ_l are the right and left torques acting in the motors, M_T is the mass of the frame of the mobile entity. $\ddot{\psi}_r$ and $\ddot{\psi}_l$ are the right and left angular accelerations of the wheels. These five equations of motion can easily be put in matrix form as stated in equation 6.11.

$$M_I(q) = \begin{bmatrix} M_T + 2m_w & 0 & -2m_w dsin\theta & 0 & 0 \\ 0 & M_T + 2m_w & 2m_w dcos\theta & 0 & 0 \\ -2m_w dsin\theta & 2m_w dcos\theta & I_T + 2m_w l^2 + I_w & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix}$$

$$\ddot{q} = \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{\theta} \\ \ddot{\psi}_r \\ \ddot{\psi}_l \end{bmatrix} \qquad V(q,\dot{q}) = \begin{bmatrix} -2m_w d\dot{\theta}^2 cos\theta \\ -2m_w d\dot{\theta}^2 sin\theta \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$B(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}$$

$$A^T(q) = \begin{bmatrix} -sin\theta & -cos\theta & -cos\theta \\ cos\theta & -sin\theta & -sin\theta \\ d & -b & b \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \tag{6.26}$$

Once the dynamic equations of motion are obtained, it is of interest to obtain a mapping function f such that $f : \tau \rightarrow \dot{\eta}$, which is indeed the dynamic model of the system. In order to do the former, it is necessary to eliminate the Lagrangian multipliers. This is achieved by differentiating equation 6.4 with respect to time t and then multiply the result by the matrix S^T and noting that $S \in N(A)$, the term $S^T A^T \lambda$ vanishes in the equation. The term $S^T B$ is equal to the identity matrix I. The procedure is shown in 6.27.

$$\begin{align} M_I\ddot{q} + V &= B\tau - A^T\lambda \tag{6.27} \\ \dot{q} &= S\eta \nonumber \\ \ddot{q} &= S\dot{\eta} + \dot{S}\eta \nonumber \\ M_I\Big(S\dot{\eta} + \dot{S}\eta\Big) + V &= B\tau - A^T \nonumber \\ S^T M_I S\dot{\eta} + S^T M_I \dot{S}\eta + S^T V &= S^T B\tau - S^T A^T \lambda \nonumber \\ S^T M_I S\dot{\eta} + S^T M_I \dot{S}\eta + S^T V &= \tau \nonumber \\ f_1 &= S^T M_I S \nonumber \\ f_2 &= S^T M_I \dot{S}\eta + s^T V \nonumber \\ \dot{\eta} &= -f_1^{-1} f_2 + f_1^{-1}\tau \tag{6.28} \end{align}$$

Equation 6.28 is the dynamic equation of the system.

6.5.5 State Representation

Choosing the state space variable $x = [q^T \eta^T]^T$ and from equations 6.4 and 6.28, the kinematic and dynamic systems can be arranged into a state representation, as stated in 6.29.

$$\dot{x} = \begin{bmatrix} S\eta \\ -f_1^{-1} f_2 \end{bmatrix} + \begin{bmatrix} 0 \\ f_1^{-1} \end{bmatrix} \tau \tag{6.29}$$

Chapter 7

Control

This chapter deals with the control strategy of a differential drive wheelchair. A wheelchair 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 prooved 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, [48, 50, 58].

First at all, a mathematical formulation about input-output feedback linearization is introduced. Secondly, a study case of a differential wheelchair mobile entity is formulated.

7.1 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 nilearizable. 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 analysed by the use of the zero dynamics of the system, [59, 60]. 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 7.1.1 (Nonlinear System)*[59]*
Let a differential dynamic equation be represented as a single-input-single-output SISO system.

$$\begin{aligned}\dot{x} &= f(x) + g(x)u \qquad (7.1)\\ y &= h(x)\end{aligned}$$

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, $u \in \mathbb{R}^n$ is the control input and $y \in \mathbb{R}$ is the output. The mappings $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^{n \times p}$ are vector fields on D. And, assuming the system 7.1 has a relative degree ρ.

Definition 7.1.2 (State Transformation Vector)*[59]*
Let ρ 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 represents the external variables, such that

$$z = T(x) = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{n-\rho}(x) \\ --- \\ h(x) \\ \vdots \\ L_f^{\rho-1}h(x) \end{bmatrix} \triangleq \begin{bmatrix} \phi(x) \\ --- \\ \varphi(x) \end{bmatrix} \triangleq \begin{bmatrix} \varsigma \\ --- \\ \xi \end{bmatrix} \qquad (7.2)$$

$$(7.3)$$

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

$$\frac{\partial \phi_i}{\partial x} g(x) = 0 \; for \; 1 \leq i \leq n - \rho \qquad (7.4)$$

Definition 7.1.3 (Normal Form System)*[59]*
The relation 7.2 will bring the system 7.1 into a normal form 7.5-7.7,

$$\begin{aligned}\dot{\varsigma} &= f_0(\varsigma, \xi) &(7.5)\\ \dot{\xi} &= A_c\xi + B_c\gamma(x)[u - \alpha(x)] &(7.6)\\ y &= C_c\xi &(7.7)\end{aligned}$$

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

$$\begin{aligned}f_0(\varsigma, \xi) &= \frac{\partial \phi}{\partial x} f(x)\Big|_{x = T^{-1}(z)} &(7.8)\\ \gamma(x) &= L_g L_f^{\rho-1} &(7.9)\\ \alpha(x) &= -\frac{L_f^\rho h(x)}{L_g L_f^{\rho-1} h(x)} &(7.10)\end{aligned}$$

Definition 7.1.4 (Input-Output Linear State Feedback Control)*[59]*
The control law that will bring the external part of the normal form into a linear one is stated as follows

$$u = \alpha(x) + \beta(x)\nu \tag{7.11}$$

The equations 7.5-7.7 result in the system

$$\begin{aligned} \dot{\varsigma} &= f_0(\varsigma, \xi) && (7.12)\\ \dot{\xi} &= A_c\xi + B_c\nu && (7.13)\\ y &= C_c\xi && (7.14) \end{aligned}$$

Definition 7.1.5 (Relative degreel)*[59]*
Let ρ be the number of times the system 7.1 is continuously derivate till the output y meets the input u, resulting in the following form

$$y^{(\rho)} = L_f^{\rho}h(x) + L_gL_f^{\rho-1}h(x)u \tag{7.15}$$

Thus, a nonlinear system of the form 7.1 has a relative degree ρ, $1 \leq \rho \leq$ if

$$\begin{aligned} L_f^i h(x) &= 0, \quad i = 1, 2, \ldots, \rho - 2 && (7.16)\\ L_gL_f^i h(x) &\neq 0, \quad i = \rho - 1 && (7.17) \end{aligned}$$

The system 7.15 can be input-output linearisable by the following equation

$$u = \frac{1}{L_gL_f^{\rho-1}h(x)}\Big[- L_f^{\rho}h(x) + \nu\Big] \tag{7.18}$$

Reducing the system 7.24 into the following linear form

$$y^{(\rho)} = \nu \tag{7.19}$$

7.2 Study Case, a Differential Drive Wheelchair

In order to apply the theory stated in section 7.1 to a differential drive wheelchair, the state representation stated in equation 6.29 must be arranged in a general nonlinear form system as presented in definition 7.1.1. For that, a nonlinear feedback $\tau = f_1u + f_2$ is applied to the state representation 6.29 bringing the system to the form.

$$\begin{aligned} \dot{x} &= f(x) + g(x)u\\ y &= h(x) \end{aligned} \tag{7.20}$$

$$f(x) = \begin{bmatrix} S(q)\eta \\ 0 \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{7.21}$$

The output equations are chosen based on the number of inputs the system contains. Since there are two inputs to the system $u = [u_1, u_2]^T$, two output equations must be chosen, $y = [y_1, y_2]^T = h(q) = [h_1(q), h_2(q)]^T$. The interest in the output equations is trajectory tracking control system. Therefor, two output equations are chosen based on the coordinates of the virtual point P_l and shown in 7.22.

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = h(q) = \begin{bmatrix} h_1(q) \\ h_2(q) \end{bmatrix} = \begin{bmatrix} x_c + (d+L)cos\theta \\ y_c + (d+L)sin\theta \end{bmatrix} \tag{7.22}$$

In order to obtain the relative degree of the system, the output y is derivated till it meets the input u.

$$\begin{aligned} \dot{y} &= \Phi(q)\eta \\ \ddot{y} &= \dot{\Phi}(q)\eta + \Phi(q)u \end{aligned} \tag{7.23}$$

The output y has been derivated twice before it encountered the input u. Thereforit is verified that the relative degree of the system is $\rho = 2$.

$\Phi(q)$ is the decoupling matrix, and defined as in equation 7.24

$$\Phi(q) = J_h(q)S(q)$$

$$With$$

$$J_h = \frac{\partial h(q)}{\partial q} \in \mathbb{R}^{(n-m)\times n} \quad as\ the\ Jacobian\ Matrix$$

In order to achieve input-output linearization, a state variable transformation vector, which is a diffeomorphism, is defined as follows, [50, 61].

$$z = T(x) = \begin{bmatrix} z_1 \\ z_2 \\ --- \\ z_3 \\ z_4 \\ --- \\ z_5 \end{bmatrix} = \begin{bmatrix} h(q) \\ --- \\ L_f h(q) \\ --- \\ \tilde{h}(q) \end{bmatrix} = \begin{bmatrix} h(q) \\ --- \\ \Phi(q)\eta \\ --- \\ \tilde{h}(q) \end{bmatrix} \tag{7.24}$$

Where $\tilde{h}(q) \in \mathbb{R}^m$ is a vectorial function such that $[J_h^T, J_{\tilde{h}}^T]$ is a full rank, [50, 61]. The system under the new state variable transformation vector $T(x)$ is characterised by

$$\dot{z} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ - - - \\ \dot{z}_3 \\ \dot{z}_4 \\ - - - \\ \dot{z}_5 \end{bmatrix} = \begin{bmatrix} \frac{\partial h}{\partial q}\dot{q} \\ - - - \\ \dot{\Phi}(q)\eta + \Phi(q)u \\ - - - \\ J_{\tilde{h}} S\eta \end{bmatrix} = \begin{bmatrix} z_3 \\ z_4 \\ - - - \\ \dot{\Phi}(q)\eta + \Phi(q)u \\ - - - \\ J_{\tilde{h}} S\eta \Big(J_h S \Big)^{-1} \begin{bmatrix} z_3 \\ z_4 \end{bmatrix} \end{bmatrix} \tag{7.25}$$

The observable part of the system 7.25, $z = [\dot{z}_1, \dot{z}_2, \dot{z}_3, \dot{z}_4]^T$, is arranged in the form presented in equation 7.26.

$$\dot{z} = A_c z + B_c \Phi(q) \Big[u - \Big(- \Phi^{-1}(q)\dot{\Phi}(q)\eta \Big) \Big] \tag{7.26}$$

With

$$A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Choosing a state feedback control law of the form showed in equation 7.11, with $\alpha(q) = -\Phi^{-1}(q)\dot{\Phi}\eta$, $\beta(q) = \Phi^{-1}(q)$ and $\gamma(q) = \Phi(q)$.

$$u = -\Phi^{-1}(q)\dot{\Phi}\eta + \Phi^{-1}(q)\nu \tag{7.27}$$

Equation 7.27 brings the system 7.26 into a linear one of the form presented in equation 7.13.

$$\dot{z} = A_c z + B_c \nu \tag{7.28}$$

The kinematic and the dynamic models, the nonlinear feedback, the nonlinear state feedback control law and the linear state transformation variables are summarized from 7.29-7.36, and sketched in figure 7.1.

$$\dot{q} = S(q)\eta \tag{7.29}$$

$$\dot{\eta} = f_1^{-1} f_2 + f_1^{-1}\tau \tag{7.30}$$

$$\tau = f_1 u + f_2 \tag{7.31}$$

$$u = \dot{\Phi}^{-1}\Big(\nu - \Phi\eta \Big) \tag{7.32}$$

$$z_1 = h_1(q) \tag{7.33}$$

$$z_2 = h_2(q) \tag{7.34}$$

$$z_3 = \Phi_1(q)\eta \tag{7.35}$$

$$z_4 = \Phi_2(q)\eta \tag{7.36}$$

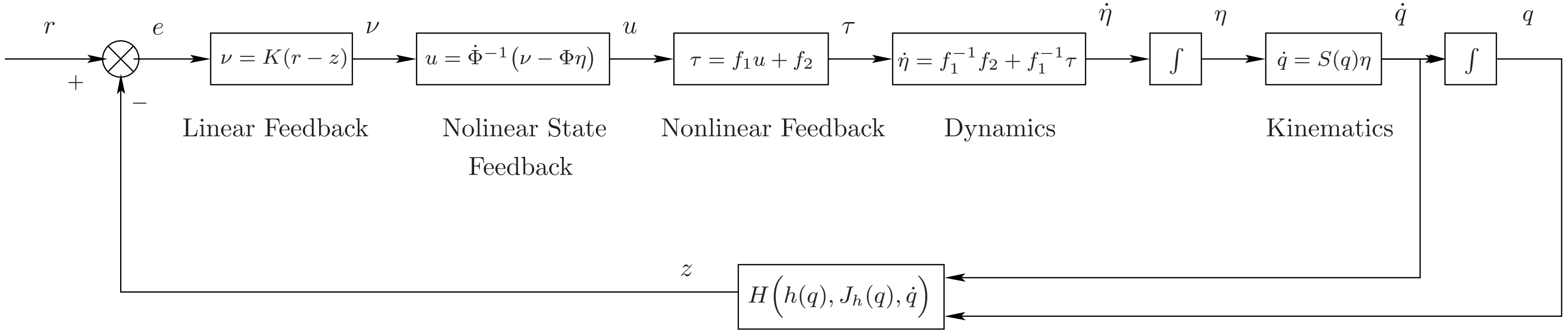


Figure 7.1: Block diagram of the control system. The lineal feedback block is in charge of placing the poles of the linear system. The nonlinear feedback linearizes and decouples the input-output map. The nonlinear feedback block allows to cancel the nonlinearities in the dynamic model. The kinematic block gives the pose of the robot in the configuration space; then the generalized coordinates are mapped, and feedback is to be compared to the reference input.

Chapter 8

Motor Control

8.1 DC Motor model

To accurately model a DC motor, an electrical as well as a mechanical part are taken into consideration, [62].

8.1.1 Electrical Part

Figure 8.1 depicts the electrical part of a DC motor

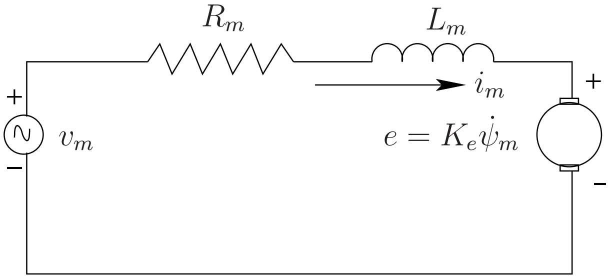


Figure 8.1: Motor electrical circuit

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

Applying Kirchoof voltage law $\sum_{k=1}^{n} v_k = 0$ to the electrical circuit in Figure 8.1 brings the electrical equation of the DC motor.

$$L_m \frac{di_m}{dt} + R_m i_m = v_m - K_e\dot{\psi}_m \tag{8.1}$$

8.1.2 Mechanical Part

Figure 8.2 depicts the mechanical part of a DC motor

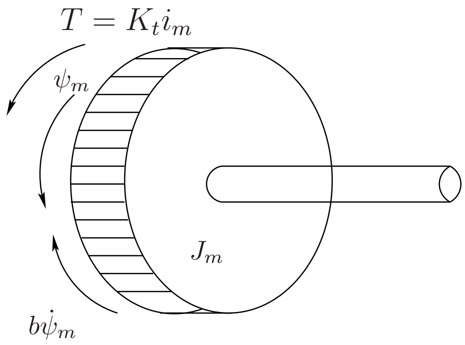


Figure 8.2: Motor mechanical part

T $[N \cdot m]$ is the motor torque, ψ $[Rad]$ is the shaft's angular position, b $[N \cdot m \cdot s]$ is the damping ratio of the mechanical system, J_m $[\frac{kg \cdot m^2}{s^2}]$ is the moment of inertia of the rotor, K_t 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 $[\frac{Rad}{s^2}]$ is the angular aceleration of the body,

$$J_m \ddot{\psi}_m + b\dot{\psi}_m = K_t i_m \tag{8.2}$$

8.1.3 Mathematical Model

Equations 8.1 and 8.2 reprsents the mathematical model of a DC motor

$$J_m \ddot{\psi}_m + b\dot{\psi}_m = K_t i_m \tag{8.3}$$

$$L_m \frac{d_{i_m}}{dt} + R_m i_m = v_m - K_e \dot{\psi}_m \tag{8.4}$$

The system can be arranged in state space representation.

$$\begin{bmatrix} \ddot{\psi}_m \\ \dot{i}_m \end{bmatrix} = \begin{bmatrix} -\frac{b}{J_m} & \frac{k_t}{J_m} \\ -\frac{K_e}{L_m} & -\frac{R_m}{L_m} \end{bmatrix} \begin{bmatrix} \dot{\psi}_m \\ i_m \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_m} \end{bmatrix} v_m \tag{8.5}$$

$$\begin{bmatrix} \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\psi}_m \\ i_m \end{bmatrix} \tag{8.6}$$

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 $v_m(s)$.

$$G_{ol}(s) = \frac{\dot{\psi}_m(s)}{v_m(s)} = \frac{K_t}{(L_m s + R_m)(J_m s + b_m) + K_t K_e} \tag{8.7}$$

In *SI* units, Kt is equal to Ke.

8.1.4 Control Design

In the control design of $D(s)$, the choice of proper specifications depends on the application. In this case the reference to the control loop is an angular velocity $\dot{\psi}$ depending on a trajectory some nonlinear system has to follow, as depicted in Figure 8.3.

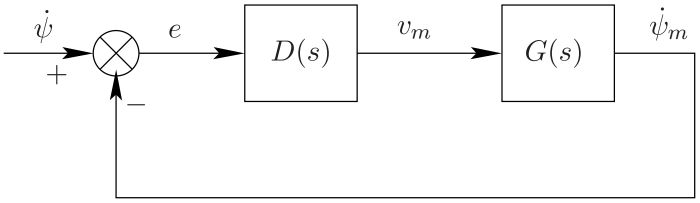


Figure 8.3: System control block

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 $\dot{\psi}_m = 0$ in equations 8.1 and 8.2

$$\frac{I_m(s)}{V_m(s)} = \frac{1}{L_m s + R_m} = \frac{1/R_m}{\tau_{ol\_fast}} \tag{8.8}$$

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 8.1 and 8.2

$$G(s) = \frac{K_m}{J_m R_m s + \left(b_m R_m + K_m^2\right)} = \frac{K_m/\left(b_m R_m + K_m^2\right)}{\tau_{ol\_slow} s + 1} \tag{8.9}$$

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.

$$s_{ol\_slow\_1} = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \tag{8.10}$$

$$s_{ol\_fast\_2} = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \tag{8.11}$$

$A = J_m L_m$, $B = -J_m R_m + L_m b_m$ and $C = b_m R_m + K_m^2$. There is no point in designing the closed loop 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 a closed loop pole $4-5$ times faster than the mechanical time constant. This will still be much (ten times) slower than the electrical constant, such that the closed loop could have one slow real and one fast pole.

For a system with a slow and a fast real poles in closed loop, the rise time tr will be almost as for a first order system $G_{ol}(s) = \frac{1}{\tau s+1}$ with output function $y(t) = 1 - exp(\frac{-t}{\tau})$.

If it is desired to have a closed loop with no steady state error, a Proportional Integral (PI) control can be used, equation 8.12. One can choose for instance the integral time constant (Ti) such that the zero of the controller cancel the slow pole of the motor and next choose the proportional constant (Kp). And, by using root locus in Matlab to give the desired dominating closed loop pole.

$$D(s) = \frac{K_p\big(T_i s + 1\big)}{T_i s} \tag{8.12}$$

A close loop transfer function is obtained by a proper combination of equations 8.12 and 8.7.

$$H(s) = G_{cl} = \frac{G_{ol} D(s)}{1 + G_{ol}(s) D(s)} \tag{8.13}$$

Section 13.2 shows the simulation results of a PI controller that has been applied to a motor transfer function $G(s)$.

Chapter 9

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 Matlab.

9.1 Nonlinear System discretization

The second order Runge-Kutter method $RK2$ can be used to solve numerically the dynamic equations of motion that govern a Nonlinear systems.

Theorem 9.1.1 (Runge-Kutter)
Let a differential dynamic equation be represented as $\dot{y}(q) = f(q)$, then $RK2$ simulates the accuracy of the second order Taylor series.

$$y_{k+1} = y_k + K_{y2} \tag{9.1}$$

$$K_{y2} = Tf\Big(t_k + \frac{T}{2}, y_k + \frac{K_{y1}}{2}\Big) \tag{9.2}$$

$$K_{y1} = Tf\big(t_k, y_k\big) \tag{9.3}$$

Where T is the sampling period, t_k is the discrete time, $f(\cdot)$ is the derivative.

Proof 9.1.1 *[63]*

The discretized nonlinear system is depicted in figure 9.1, where the dynamics of the system has been numerically solved by means of $RK2$.

$$\eta_{k+1} = \eta_k + K_{\eta 2} \tag{9.4}$$

$$K_{\eta 2} = Tf\Big(t_k + \frac{T}{2}, \eta_k + \frac{K_{\eta 1}}{2}\Big) \tag{9.5}$$

$$K_{\eta 1} = Tf\big(t_k, \eta_k\big) \tag{9.6}$$

$$q_{k+1} = q_k + K_{q2} \tag{9.7}$$

$$K_{q2} = Tf\Big(t_k + \frac{T}{2}, q_k + \frac{K_{q1}}{2}\Big) \tag{9.8}$$

$$K_{q1} = Tf\big(t_k, q_k\big) \tag{9.9}$$

9.2 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, [62].

Figure 9.2 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_k 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.

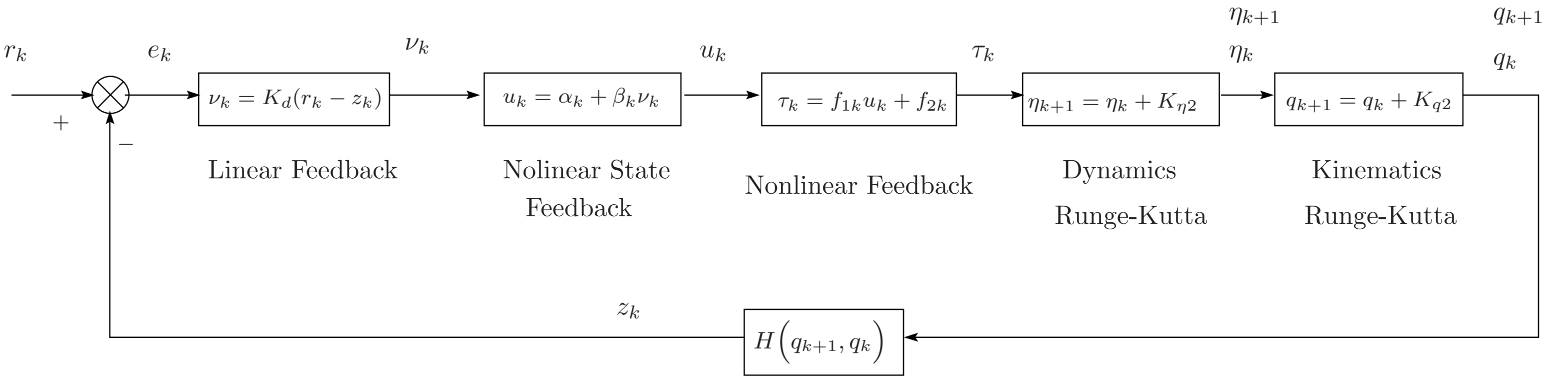


Figure 9.1: Discrete nonlinear system block diagram

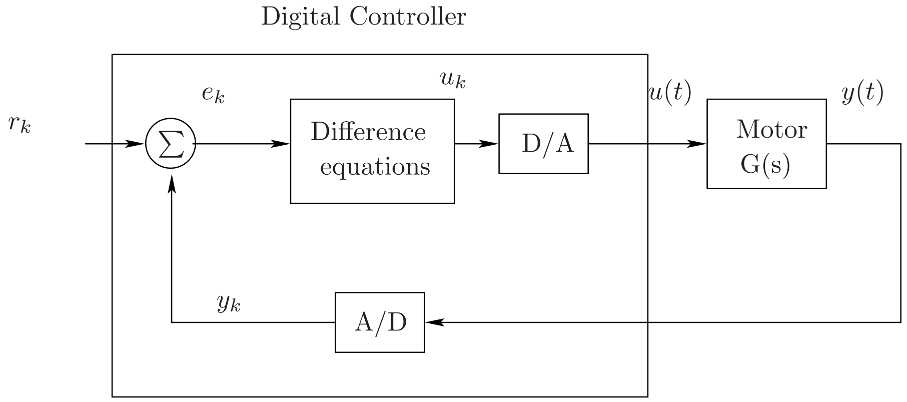


Figure 9.2: Digital block diagram controller

Chapter 10

System

The complete system, composed of the modules described in previous sections, is depicted in Figure 10.1. The map making block is responsible for generating a map of the environment based on laser readings. The path planning block has to generate a smooth path from the start to the goal, which means that the coordinate must be chosen either by the BCI or tongue interfaces. Once the path is generated, it is used as a reference that is compared with the output from the system generating an error that will be used as the input to the control. The localization block together with the output from the actuators are used to generate the output signals that have to be compared with the reference input. The control block, which dynamic equations have been discretized by means of second order Runge-Kutta method [63], generates the control signals to the actuators that will follow a specific path.

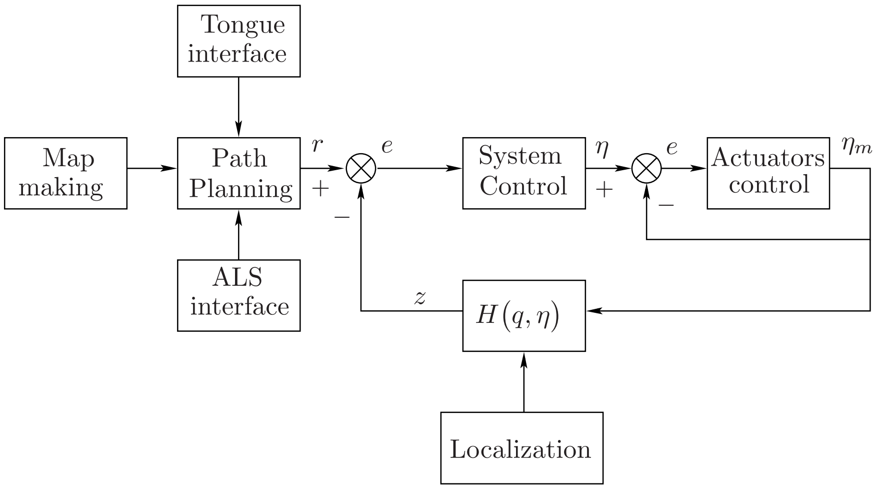


Figure 10.1: Block diagram of the complete system. The path planning is achieved after a point destination is selected on the map. This action produces a reference input to the control module. Then the control path tracking strategy is applied to give the right commands to the actuators.

Part III

Simulation Results

Chapter 11

Localization

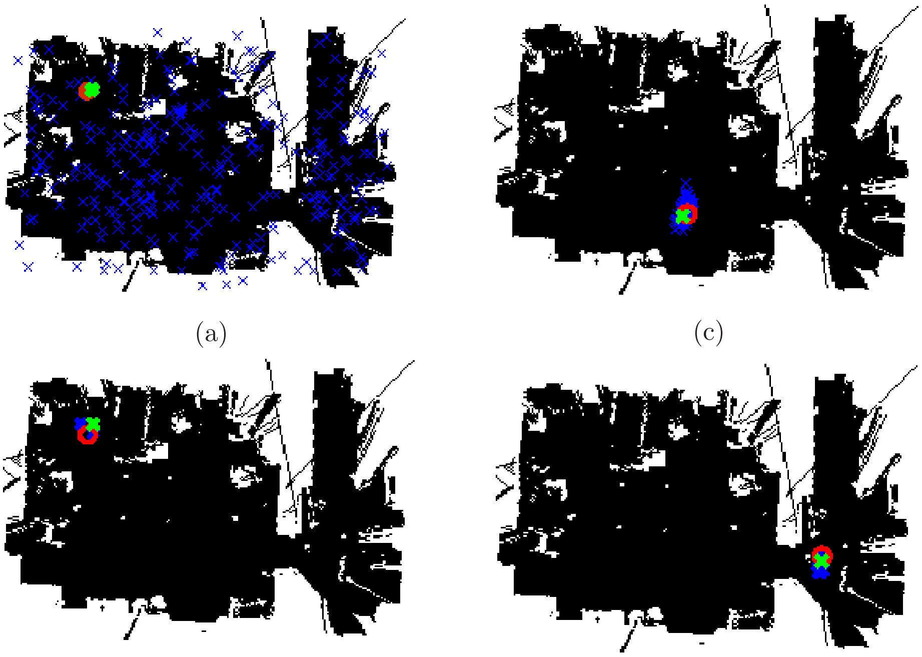
For the first laser sampling, three iterations of the KLD algorithm were performed, while only one iteration was performed for the remaining laser samplings as shown in Figure 11.1.

- Figure 11.1(a) shows the result of the algorithm in the firsts laser measurement after one iteration. The red circle is the true pose, the green cross is the best estimate pose calculated by the algorithm, and the blue crosses are random particles that remain after the first iteration.
- Figure 11.1(b) shows the result of the algorithm that is still in the first laser measurement, e.g. the mobile entity has not moved at all. The algorithm has performed three iterations, where random particles have been eliminated, and the ones that remain are concentrated around the best estimate.
- Figures 11.1(c) and (d) show the results of the algorithm for the 10^{th} and 20^{th} laser samplings after one iteration.

It can be seen that the KLD algorithm accurately tracks and localizes the mobile entity in the configuration space of the map as shown in Figure 11.2. Table 11.1 shows the true poses and the best estimates in $[x, y]$ that correspond to the previous measurements.

Table 11.1: Comparison between true pose and best estimate using KLD

	(x, y) cm		
Position Samplings	True pose	Best estimate	Error (module)
1 (1 iteration)	$(225, 250)$	$(220, 265)$	15.8
1 (3 iterations)	$(225, 250)$	$(195, 265)$	33.5
10 (1 iteration)	$(585, 540)$	$(590.3, 527.8)$	13.2
20 (1 iteration)	$(540, 910)$	$(553.8, 908.8)$	13.6



(a)

(c)

(b)

(d)

Figure 11.1: (a) shows the result of the algorithm for the first laser sampling after 1 iteration. (b) result of the algorithm for the first laser sampling after 3 iterations. (c) result of the algorithm for the 10^{th} laser sampling after 1 iterations. (d) result of the algorithm for the 20^{th} iteration after 1 iterations.

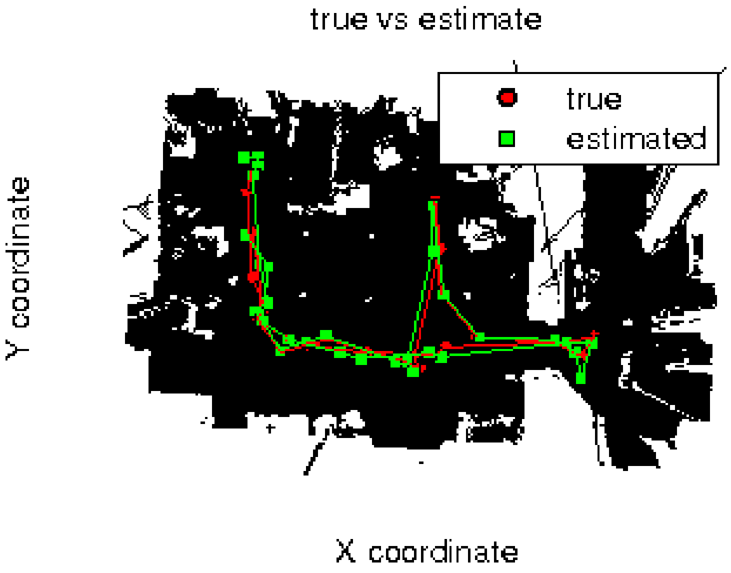


Figure 11.2: Comparison between the true poses and best estimates.

Chapter 12

Motion Planning

This chapter addresses motion planning simulation based on the solution of Laplace's equation. The method was explained in section 5.4. Figure 12.1(a) shows a map of a laboratory based on laser readings. The black area represents free space (C_{free}) while the white area represents occupied space or C-obstacle-region (CB_{region}). Figure 12.1(b) shows the configuration space based on the morphological operation called dilation; the map has been dilated a factor of 20 cm. Dilation was explained in chapter 5.5.

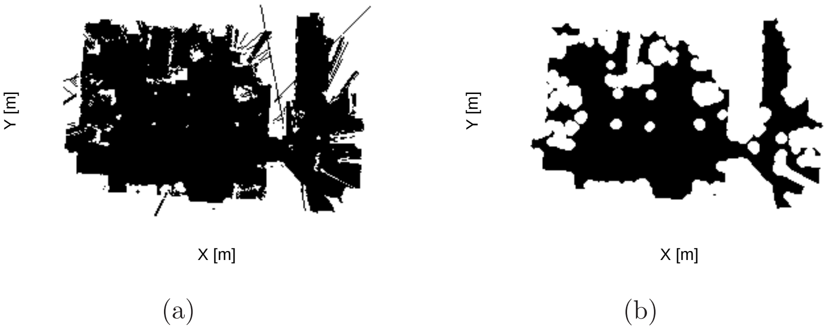


Figure 12.1: (a) Map created from laser readings. (b) Configuration space created by dilation.

Figure 12.2(a) depicts a start point represented by a blue circle and a goal configuration represented by a red square. The planner has to find a smooth path from start to goal configuration, based on the solution of Laplace's equation. The smooth path can be depicted in figure 12.2(b). Figure 12.3 shows a 3D representation of the solution of Laplace's equation applied to the map from figure 12.1(b).

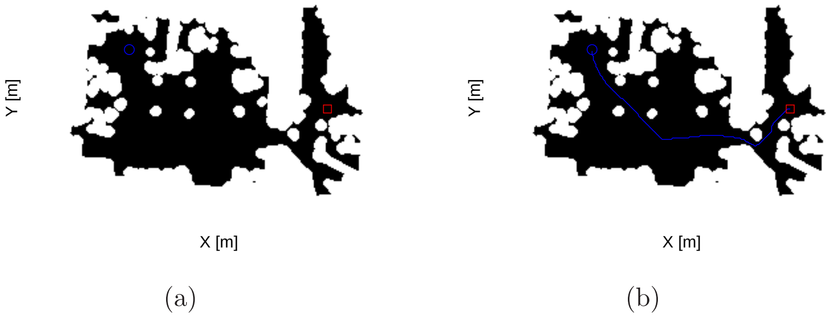


Figure 12.2: (a) The blue circle represents the start point configuration. (b) The red square represents the the goal configuration. (b) Shows a smooth path (τ) in the free configuration space (C_{free}).

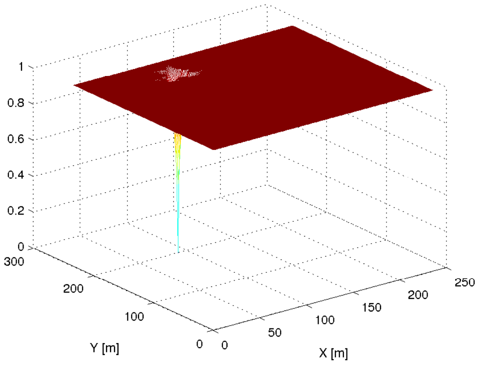


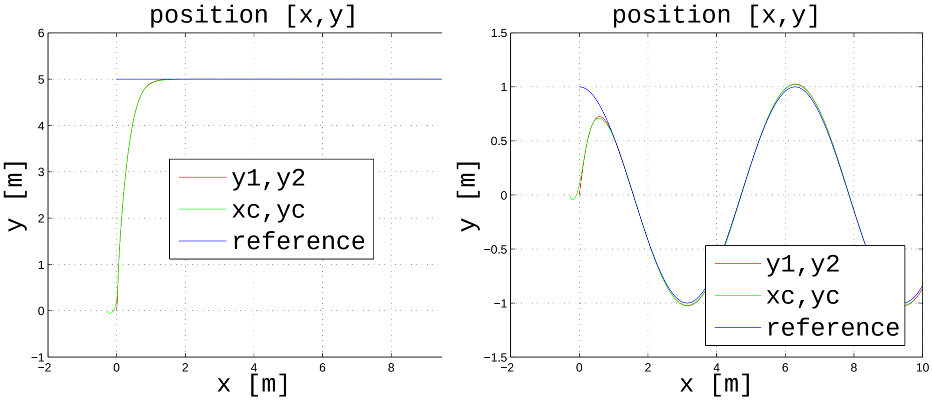
Figure 12.3: A 3D representation of the solution of Laplace's equation. The peak represents the goal configuration.

Chapter 13

Control

13.1 Nonlinear Control

The input-output state feedback nonlinear control path following for a constant and a cosine reference paths are depicted in figures 13.1(a)(b) and 13.2(c)(d). They show and exponential and linear path tracking, where y_1, y_2 represents the coordinates of the virtual point P_l and x_c, y_c represents the coordinates of the center of mass.



(a)

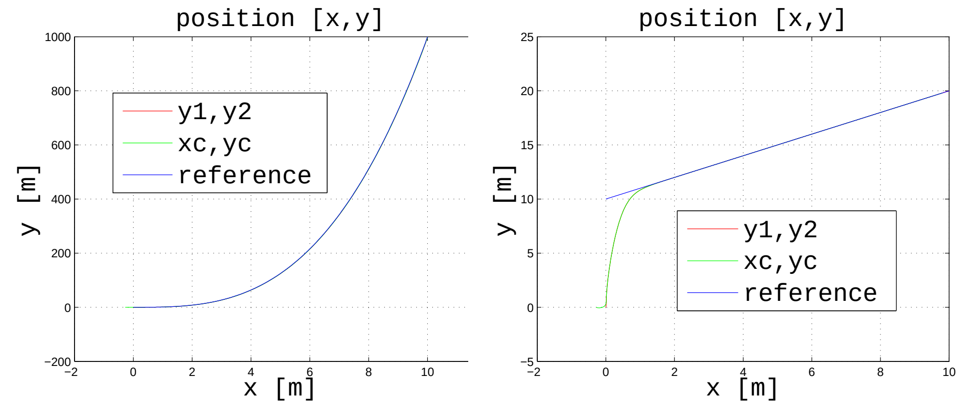
(b)

Figure 13.1: (a) Constant path tracking. (b) Cosine path tracking.

13.2 Motor Control

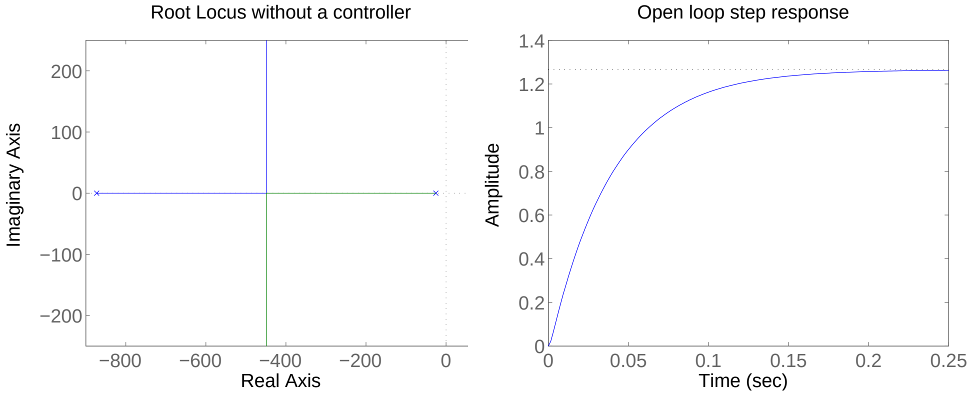
The motor parameters used in the simulation are; $J_m = 0.0103$, $b_m = 0.0034$, $Km = 0.78$, $R_m = 2.4242$, and $L_m = 0.0027$. Figure 13.3(a) shows the slow and fast poles of the system $G(s)$, which values are $s_{ol\_slow\_1} = -25.4056$ and $s_{ol\_fast\_2} = -872.7764$.

Figure 13.3(b) depicts the step response of the open loop system. It can be seen there is a steady state error.



(c) (d)

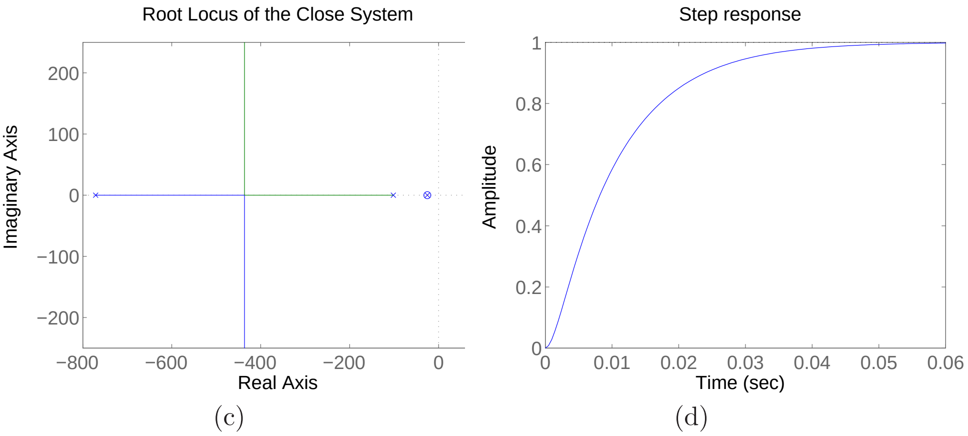
Figure 13.2: (c) Exponential tracking. (b) Line tracking.



(a) (b)

Figure 13.3: (a) Open loop poles. (b) Open loop step function.

Figure 13.4(c) depicts the close loop poles of $G_{cl}(s)$. The integral constant TI in equation 8.12 has been chosen to cancel the slow open loop pole in order for the entire system to behave as a second order one. The slow close loop pole has been chosen to be around 4 times faster than the slow open loop pole using the commands in Matlab rlocus and rlocfind. Finally, a step response of the close loop system can be seen in Figure 13.4(d).



(c) (d)

Figure 13.4: (d) Close loop poles. (d). Close loop step response.

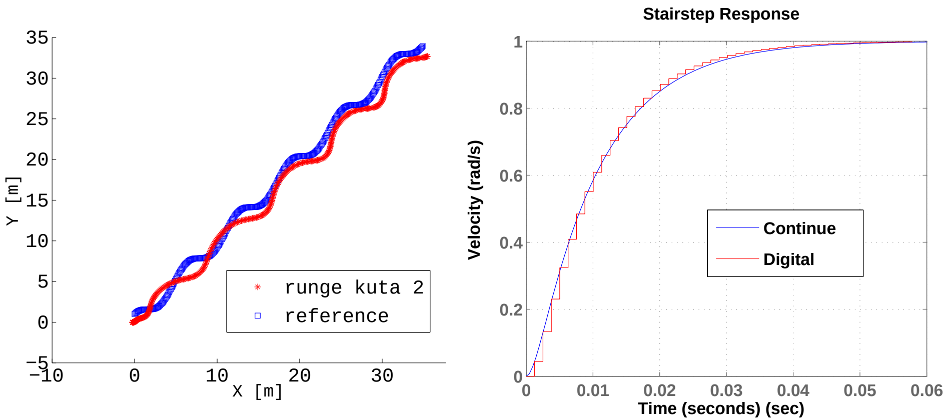
The rise time t_r and the settling time t_s of the system can be obtained as follows

$$t_r = 2.2\tau_{cl\_slow} \quad (13.1)$$
$$t_s = 3.9\tau_{cl\_slow} \quad (13.2)$$

Where τ_{cl_slow} from Figure 13.4(c) has the value of 0.00981 making $t_r = 0.021\ sec$ and $t_s = 0.038\ sec$

13.3 Digital Control

Figure 13.5(a) depicts the simulation results of the nonlinear system trajectory tracking using $RK2$. Figure 13.5(b) presents a digital step response that corresponds to a close loop system with open loop transfer function $G_{ol}(z)$ and a Digital compensator $D(z)$



(a)

(b)

Figure 13.5: (a) Discrete nonlinear trajectory tracking. (b) Digital step response.

Chapter 14

System Simulation

The simulation results of the individual blocks of the system have been shown in the previous chapters. The main objective of this section is to show the control simulation result of the system, as depicted it in Figure 14.1. The nonlinear control path following scheme is presented as a dash line, and the reference input is represented as a solid line.

The Mean Square Error between the reference input and the control tracking ($MSEre$) for the simulation performed in Matlab is 6 cm^2.

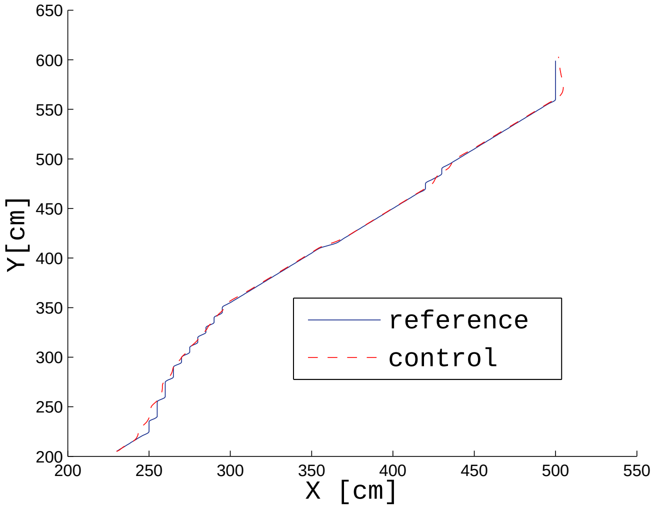


Figure 14.1: Control simulation. The reference input is represented as a solid line, and the control trajectory tracking is represented as a dash line.

Part IV

Summary, Conclusion and Future Work

A differential mobile robot based wheelchair has been described and simulated in this report. To this end, modules for sensor fusion, map making, destination point selection, localization, path planing and control strategy were introduced.

The uncertainty inherited in the sensor's readings were taken into consideration by the use of probabilistic sensor models, and the Bayesian method was used to update the probabilistic map into an occupancy grid. It has been shown that the Kullback Leibler Divergence localization algorithm localized, with great accuracy, the wheelchair in its configuration space. For destination point selection, BCI is proposed. A nonlinear control strategy for the autonomous wheelchair was selected based on the high nonlinearities of the system, focusing mainly on state feedback input-output linearization. This process has the property of splitting the system into external and internal parts, making the external part suitable for linearization. Boundary internal part stability conditions were analyzed using the zero dynamics of the system. For trajectory tracking, two external outputs were selected and a linear feedback loop was designed, in order to achieve achieves good results in the path trajectory tracking. The dynamics of the system were analyzed based on the Lagrangian formalism to holonimic-nonholonomic systems.

Simulation results of each individual block as well as the system were presented to illustrate a high control wheelchair based on a destination point selection.

It can be observed in Figure 11.1(b) that the KLD considerably reduces the number of particles during the localization tracking after 3 iterations during the first laser measurement. Table 11.1 can be appreciated that the best pose estimation for the first laser sampling, that represents the wheelchair in still position (initialization), has an accuracy error of 15.8 cm and 33.5 cm after 3 iterations. For the 10^th and 20^th laser samplings, that represent position of the wheelchair in movement, the accuracy errors are 13.2 cm and 13.6 cm respectively.
Statistical analysis of the trajectory control tracking of the system Figure 14.1 showed mean squared error of 6 cm^2, which ensures accurate navigation due to the 20cm dilation of the occupied spaces.

The simulation results in this paper have shown the feasibility of using the proposed system, based on a single command from the user, to provide augmentative mobilization to individuals with severe physical impairments, such as high-level spinal cord injury or advanced states of amyotrophic lateral sclerosis. Such persons could use e.g. BCI [69], in order to successfully choose destination points.

Finally, a differential mobile robot wheelchair prototype has been built from scratch with the purpose of testing the aforementioned algorithms. Then, on success these algorithms need to be transferred to a real wheelchair.

Bibliography

[1] R.F. Edlich, K.P. Nelson, M.L. Foley, R.M. Buschbacher, W.B. Long and E.K. Ma, "Technological advances in powered wheelchairs", *J. Long. Term. Eff. Med. Implants*, vol. 14, pp. 107-130, 2004.

[2] H.V. Christensen and J.C. Garcia, "Infrared Non-Contact Head Sensor, for Control of Wheelchair Movements", *Assistive Technology: From Virtuality to Reality, A.Pruski and H.Knops (Eds) IOS Press*, pp. 336-340, 2003.

[3] S. Guo, R.A. Cooper, M.L. Boninger, A. Kwarciak and B. Ammer, "Development of power wheelchair chin-operated force-sensing joystick", *in Proceedings of the 2002 IEEE Engineering in Medicine and Biology 24th Annual Conference and the 2002 Fall Meeting of the Biomedical Engineering Society (BMES / EMBS), Houston, TX*, 2002, pp. 2373-2374.

[4] W. Nutt, C. Arlanch, S. Nigg and G. Gtaufert, "Tongue-mouse for quadriplegics", *J Micromech Microengineering*, vol. 8, pp. 155, 1998.

[5] X. Huo and M. Ghovanloo, "Evaluation of a wireless wearable tongue-computer interface by individuals with high-level spinal cord injuries",*J. Neural Eng.*, vol. 7, 2010.

[6] M.E. Lund, H.V. Christensen, H.A. Caltenco Arciniega, E.R. Lontis, B. Bentsen and L.N.S. Andreasen Struijk, "Inductive tongue control of powered wheelchairs", *in Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Buenos Aires, Argentina*, 2010, pp. 3361-3364.

[7] J.S. Agustin, J.C. Mateo, J.P. Hansen and A. Villanueva, "Evaluation of the Potential of Gaze Input for Game Interaction", *PsychNology Journal*, vol. 7, pp. 213-236, 2009.

[8] A.F. Cabrera and K. Dremstrup, "Auditory and spatial navigation imagery in Brain-Computer Interface using optimized wavelets", *J. Neurosci. Methods*, vol. 174, pp. 135-146, 9/15, 2008.

[9] T. Felzer and R. Nordmann, "Alternative wheelchair control", *Proc.RAT*, vol. 7, pp. 67-74, 2007.

[10] L.N.S. Andreasen Struijk, "An inductive tongue computer interface for control of computers and assistive devices", *IEEE Trans. Biomed. Eng.*, vol. 53, pp. 2594-2597, DEC, 2006.

[11] L.N.S. Andreasen Struijk, E.R. Lontis, B. Bentsen, H.V. Christensen, H.A. Caltenco Arciniega and M.E. Lund, “Fully integrated wireless inductive tongue computer interface for disabled people”, *in Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, USA.*, 2009, pp. 547-550.

[12] A. K. Mitra “Finite Difference Method for the Solution of Laplace Equation”, *Department of Aerospace Engineering, Iowa State University*

[13] C.I. Connolly, J.B. Burns, R. Weiss “Path Planning Using Laplace’s Equation”, *In Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 2102–2106, 1990,

[14] C.I. Connolly, R.A. Grupen “The application of Harmonic Functions to Robotics”, *Journal of Robotics Systems*, pages 931–946, Vol. 10, No. 7, 1993,

[15] C.I. Connolly “Harmonic Functions and Collision Probabilities”, *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3015–3019, 1994,

[16] P. Štepán and L. Přeučil and M. Kulich “Robust Data Fusion with Occupancy Grid”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, pages 106–115, Vol. 35, No. 1, 2005,

[17] J. Barraquand, B. Langlois and J.C. Latombe “Numerical Potential Field Techniques for Robot Path Planning”, *IEEE Transactions on Systems, Man, and Cybernetics*, pages 224–240, Vol. 22, No. 2, 1992,

[18] L.C. Lai, C.J. Wu and Y.L. Shiue “A Potential Field Method for Robot Motion Planning in Unknown Environments ”, *Journal of the Chinese Institute of Engineers*, pages 369–377, Vol. 30, No. 3, 2007,

[19] O. Khatib “Real Time Obstacle Avoidance for Manipulators and Mobile Robots”, *International Journal of Robotics Research*, pages 90–98, Vol. 5, No. 1, 1986,

[20] D.F. Guang, J. Peng, B. Xin; W. Hong “ AUV local path planning based on virtual potential field”, *IEEE International Conference, Mechatronics and Automation*, pages 1711–1716, Vol. 4, 2005,

[21] H. Choset, K.M.. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.E. Kavraki and S. Thrun “ Principles of Robot Motion, Theory, Algorithms and Implementations”, *The MIT Press, Cambridge, Massachusetts, London, England*, 2005,

[22] J.C. Latombe “ Robot Motion Planning”, *Kluwer Academic Publications*, 1991,

[23] R. Courant and D. Hilbert “ Methods of Mathematical Physics I”, *New York: Interscience*, 1953,

[24] C.I. Connolly “Applications of Harmonic Functions to Robotics”, *Journal of Robotic Systems*, 1992,

[25] G. Dudek and M. Jenkin “Computational Principles of Mobile Robotics”, *Cambridge University Press*, 2000,

[26] R.C. Gonzalez and R.E. Richard “Digital Image Processing”, *Prentice Hall*, 2002,

[27] F. Dellaert, D. Fox, W. Burgard and S. Thrun “ Monte Carlo Localization for Mobile Robots”, *IEEE International Conference on Robotics and Automation*, 1999,

[28] D. Fox, W. Burgard, F. Dellaert, and S. Thrun “ Monte Carlo Localization : Efficient Position for Mobile Robots”, *In Proc. of the National Conference on Artifial Intelligence*, 1999,

[29] D. Fox “Adapting the Sample Size in Particle Filters Through KLD-Sampling”, *International Journal of Robotics Research*, Vol. 22, 2003

[30] R. Ioannis “A particle filter tutorial for Mobile Robot Localization”, *Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University*, 2004

[31] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp “A tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking”, *IEEE Transactions on Signal Processing*, pages 174–188, Vol. 50, No. 2, 2002

[32] S. Thrun, D. Fox, W. Burgard and F. Dellaert, “Robust Montecarlo Localization for Mobile Robots”, *Artificial Intelligence*, pages 99–141, Vol. 128, 2001,

[33] S. Thrun, D. Fox and W. Burgard “Probabilistic Robotics”, *The MITT Press*, 2006

[34] S. Keith, “A Guide to Installing RTAI Linux ”, *Anglo-Australian Observatory*, 2004

[35] C. Sousa, “How-to Install RTAI in Ubuntu Hardy”, 2008

[36] C. Sousa, “RTAI Tutorial”, *Combra University*, 2009

[37] J. Monteiro, “RTAI Installation Complete Guide”, 2008

[38] K. Sam Shanmugan and A.M. Breipohl “Random Signals Detection Estimation and Data Analysis”, *Wiley*, 1988,

[39] L. A. Klein “Sensor and Data Fusion, A Tool for Information assessment and Decision making”, *SPIE-The International Society for Optical Engineering,Bellingham, Washington USA*, 2004,

[40] E. P. Box and C. Tiao “Bayesian Inference In Statistical Analysis”, *Addison Wesley*, 1973,

[41] S.J. Henkind and M.C. Harrison “An Analysis of Four Uncertainty Calculi”, *IEEE Transactions on Systems, Man and Cybernetics* , pages 700–714, Vol. 18, No. 5, 1988

[42] Ronald E. Walpole “Probability and Statistics For Engineers and Scientists”, *Prentice Hall*, 1997

[43] J. Borenstein and Y. Koren “The vector field Histogram-Fast Obstacle Avoidance for Mobile Robots”, *IEEE Transactions on Robotics and Automation*, pages 278–288, Vol. 7, No. 3, 1991

[44] A. Elfes “Sonar-Based Real-World Mapping and Navigation”, *IEEE Journal on Robotics and Automatation*, pages 249–265, Vol. 3, 1987

[45] H.P. Moravec “Sensor Fusion in Certainty Grids for Mobile Robots”, *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pages 61–74, Vol. 9, No. 2, 1988

[46] H. Moravec and A. Elfes “High Resolution Maps from Wide Angle Sonar”, *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pages 116–21, 1985

[47] K. R. Symon “Mechanics, third edition”, *Addison-Wesley*, 1972

[48] P.Coelho and Urbano Nunes “Lie Algebra application to mobile robot control: a tutorial”, *Robotica*, pages 483–493, Vol. 21, 2003

[49] C.Campion, G.Bastin and B.D. Novel “Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots”, *IEEE Transactions on Robotics and Automation*, pages 47–62, Vol. 2, No, 1, 1996

[50] N.Sarkar, N.Yung and R.V.Kumar “Control of Mechanical Systems with Rolling Constraints: Application to Dynamic Control of Mobile Robots”, *Technical Report, Department of Computer* & *Information Science (CIS), University of Pensylvania*, 1992

[51] A.M. Bloch “Nonholonomic Mechanics and Control”, *Springer, ISBN 0-387-95535-6*, 2000

[52] H.Goldstain “Classical Mechanics”, *Addison-Wesley, 2 edition, ISBN 0-201-02918-9*, 1980

[53] D. Ding, A. Copper and S. Guo and T.A. “Analysis of Driving Backward in an Electric-Power Wheelchair”, *IEEE Transactions on Control Systems Technology*, pages 934943, Vol. 12, No. 6, 2004

[54] B.W. Jhonson, J.H. Jhonson “Dynamic Modeling of an Electric Wheelchair”, *IEEE IEEE Transactions on Industry applications*, pages 12841293, Vol. 1A-21, No. 5, 1985

[55] X.Q. Chen, J.G. Chase, P. Wolm “System Identification and Modelling of Front Drive Electric Wheelchairs”, *17th International Federation of Automatic Control World Congress*, 2008

[56] T.J.A. de Vries, C. van Heteren and L. Huttenhuis “Modelling and control of a fast moving, highly maneuverable wheelchair”, *Proc. of the International Biomechatronics Workshop*, pages 110115, 1999

[57] G. Campion, G.Bastin and B.DAndrea-Nove "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots", *Proceedings of 1993 IEEE International Conference on Robotics and Automation*, pages 462469, vol. 1, 1993

[58] H.Nijmeijer, A.J.van der Schaft "Nonlinear Dynamic Control Systems", *Springer-Verlag, New York*, 1990

[59] H.Khalil "Nonlinear Systems" *Prentice Hall, ISBN 0-13-067389-7*, 2002

[60] H.J.Marquez "Nonholonomic Control Systems Analysis and Design", *Wiley, ISBN 0-471-42799-3*, 2003

[61] X. Yun, N.Sarkar, R.V.Kumar and E. Pljug "Control of Multiple Arm Systems with Rolling Constraints ", *Technical Report, Department of Computer & Information Science (CIS), University of Pensylvania*, 1991

[62] G.F.Franklin and J.D.Powell "Feedback Control of Dynamic Systems, third edition", *Addison-Wesley, ISBN 0-201-53487-8*, 1994.

[63] J.M.Mathews "Numerical Methods for Mathematics, Science, and Engineering, second edition", *Prentice Hall , ISBN 0-13-624990-6*, 1992.

[64] P. Moravec and A. Elfes "High Resolution Maps from Wide Angle Sonar", *IEEE Journal of Robotics and Automation*, pp. 116–121, 1985.

[65] A. Elfes "Using Occupancy Grids for Mobile Robot Perception and Navigation", *IEEE Journal of Robotics and Automation*, Vol. 22, pp. 45–57,1989.

[66] A. Elfes "A Tesselated Probabilistic Representation for Spatial Perception and Navigation", *Proceedings of the NASA Conference on Space Telerobotics*, pages 341–350, vol. N90, 1989.

[67] A. Elfes and L. Matthies "Integration of Sonar and Stereo Range Data Using a Grid-Based Representation", *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 727–733, 1988.

[68] A. Elfes and L. Matthies "Sensor Integration for Robot Navigation: Combining Sonar and Stereo Range Data in a Grid-Based Representation", *Proceedings of the 26th IEEE Conference on Decision and Control*, address=Los Alamitos, Calif., 1987.

[69] A.R. Cabrera and K. Dremstrup "Steady-State Visual Evoked Potentials to drive a Brain Computer Interface", *Aalborg University. Department of Health Science and Technology Aalborg, ISBN 8790562712*, 2008

Prepared in LaTeX 2_ε

Appendix A

Proofs

A.1 Bayes' rule

Theorem A.1.1 (Total probability theorem)
B_1, $B2$,...,B_k belongs to the sample space S such that $P(B_i \neq 0)$ for $i = 1, 2, ..., k$, then, for every event A of S,

$$P(A) = \sum_{i=1}^{k} P(B_j \cap A) = \sum_{j=1}^{k} P(B_j)P(A|B_j) \tag{A.1}$$

Proof A.1.1 (Total probability theorem)
From figure A.1, definition A.1.1 and corollary A.1.1.

$$\begin{aligned} A =& (B_1 \cap A) \cup (B_2 \cap A) \cup \cdots \cup (B_k \cap A) \\ P(A) =& P(B_1 \cap A) + P(B_2 \cap A) + \cdots + P(B_k \cap A) \\ P(A) =& \sum_{j=1}^{k} P(B_j \cap A) = \sum_{j=1}^{k} P(B_j)P(A|B_j) \end{aligned} \tag{A.2}$$

Proof A.1.2 (Bayes' rule) *Equation 3.1.1.*

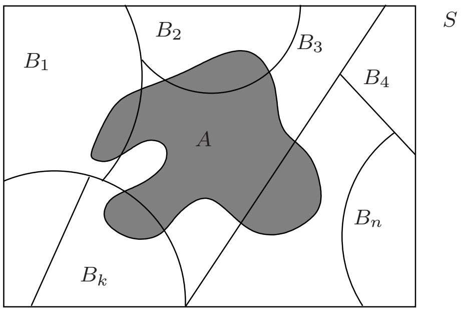


Figure A.1: Partition of the sample space S.

From the definition of conditional probability A.1.1 and the total theorem A.1.1,

$$
\begin{aligned}
P(B_i|A) =& \frac{P(B_i \cap A)}{P(A)} \\
P(B_i|A) =& \frac{P(B_i \cap A)}{\sum_{j=1}^{k} P(B_j)P(A|B_j)} \\
P(B_i|A) =& \frac{P(B_i \cap A)}{\sum_{j=1}^{k} P(B_j \cap A)} = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{k} P(B_j)P(A|B_j)} \qquad \text{(A.3)}
\end{aligned}
$$

Definition A.1.1 (Conditional probability)
$P(B|A)$ represents the conditional probability of an event B given an event A and it is defined as follows

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad \text{if} \quad P(A) > 0 \qquad \text{(A.4)}$$

Definition A.1.2 (Independent events)
Two events, A and B, are independent if and only if

$$
\begin{aligned}
P(A \cap B) &= P(A)P(B) \\
P(B \cap A) &= P(B)P(A) \\
P(B|A) &= P(B) \\
P(A|B) &= P(A) \qquad \text{(A.5)}
\end{aligned}
$$

Corollary A.1.1 *If the events A_1, A_1, A_2, A_3, A_n are mutually exclusive then*

$$P(A_1 \cup A_2 \cup A_3 \cup \cdots \cup A_n) = P(A_1) + P(A_2) + P(A_3) + \cdots + P(A_n) \quad \text{(A.6)}$$

Appendix B

RTAI Instalation Guide

This appendix describes RTAI-3.6.1 installation on Debian with Linux kernel 2.6.24. [34, 35, 36, 37]

B.1 Introduction

The instalation is done on EPIA M10000 Mini ITX Mainboard running Debian with Linux kernel 2.6.26

Appendix C

Schematic Motor Interfaces

Figures C.1 and C.2 shows the schematic diagras of the input output interfaces respectively.

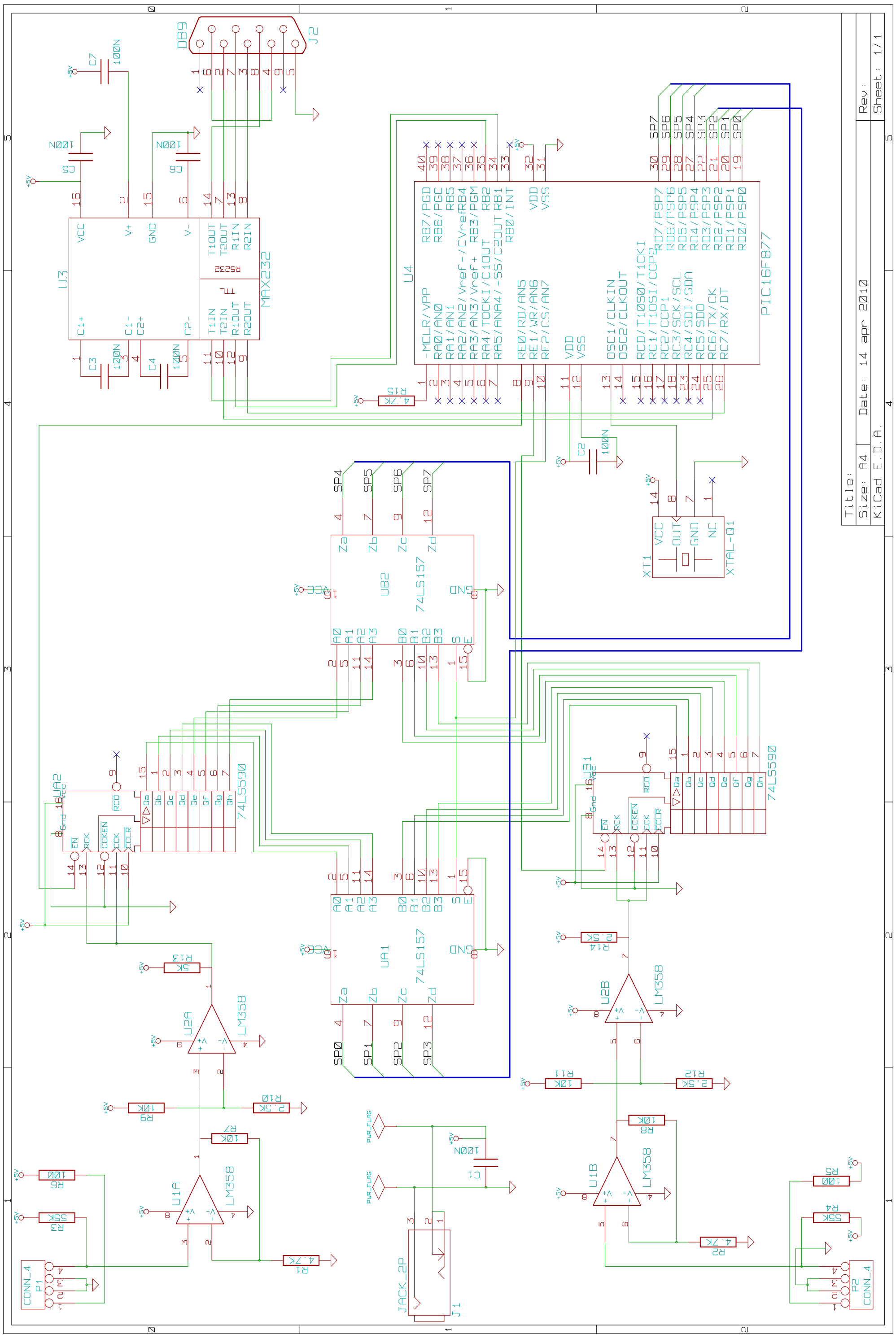


Figure C.1: Schematic of the output inteface.

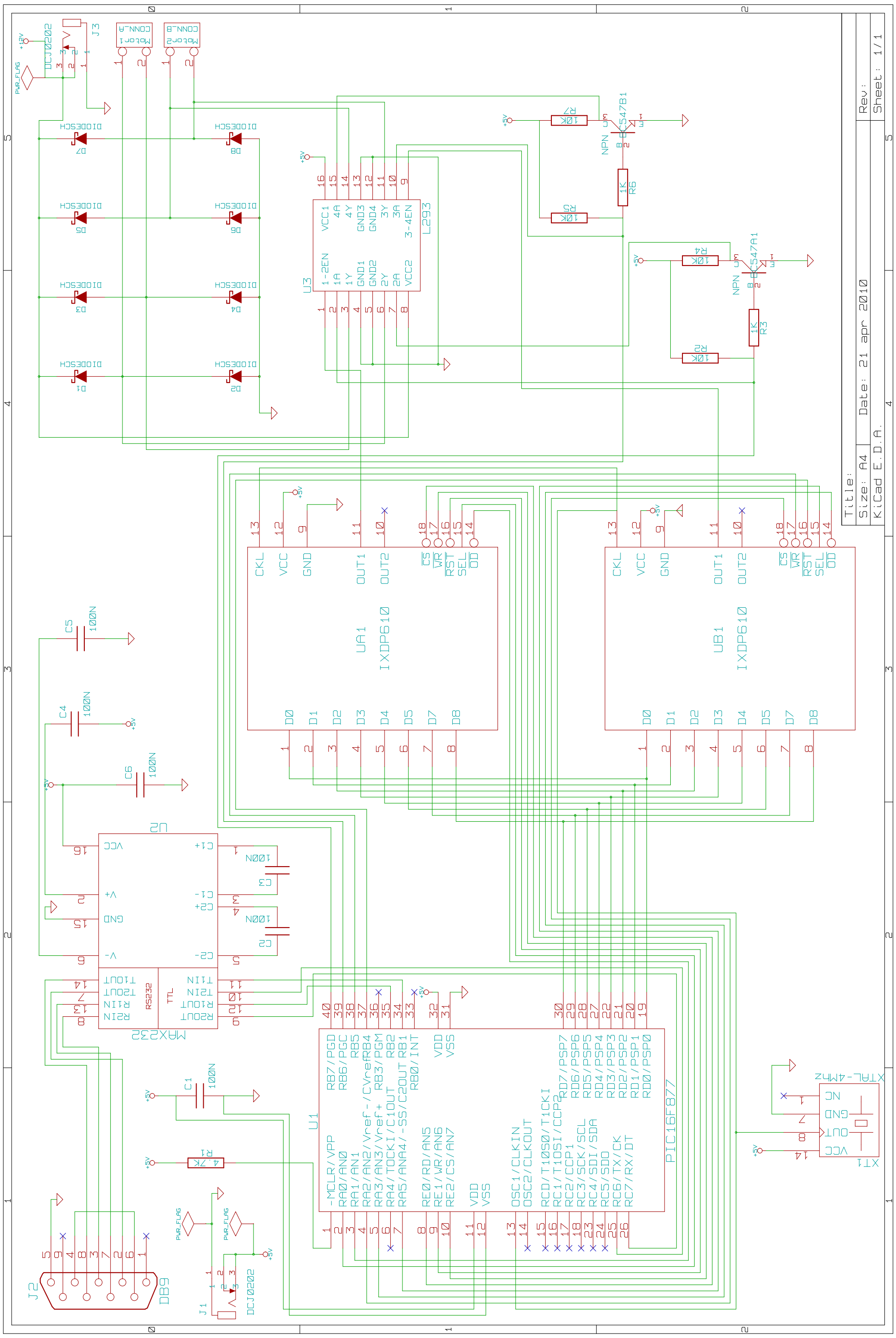


Figure C.2: Schematic of the input interface.