

UNIVERSITY OF SZEGED
Faculty of Science and Informatics

Faculty of Informatics

Department of Technical Informatics

Doctoral School of Computer Science

**Optimizing robotic actuator, trajectory tracking
and fuzzy control**

PhD Thesis

Tamás Szépe

Supervisor

Dr. Gyula Mester

Szeged, 2015

Contents

1	Introduction	1
1.1	Summary of the Author's Contributions	2
2	Pneumatic Artificial Muscle Actuator	3
2.1	Introduction	3
2.2	Problem Statement	5
2.3	The Transfer Function	6
2.4	Experimental Setup	7
2.4.1	Model Prediction Capabilities	9
2.4.2	Calibration Procedure	12
2.5	Benefits of Using the Developed Transfer Function	14
2.6	Using PAMs as actuators	15
2.7	Designing Principals of the Simulator	17
2.8	Implementation of the Numerical Simulation Constraints	19
2.9	Experimental Results	22
2.9.1	Design Optimization with Genetic Algorithm	22
2.9.2	Validation of the Design Using an Experimental Device	24
2.10	Benefits of Using the Spool and Cable Transmission System	26
3	Pure Pursuit Trajectory Tracking	28
3.1	Introduction	28
3.2	Optimal Trajectory Planning Algorithm	31

CONTENTS

3.3	Trajectory Tracking Controllers	32
3.3.1	Feedforward Controller	32
3.3.2	Linear State Tracking Controller	33
3.3.3	Nonlinear State Tracking Controller	33
3.4	Pure Pursuit Trajectory Tracking Approach	34
3.4.1	Review of the Basic Pure Pursuit Path Tracking	34
3.4.2	The Developed Pure Pursuit Trajectory Tracking Approach	36
3.5	Experimental Validation and Comparative Study	39
3.6	Benefits of Pure Pursuit Trajectory Tracking	44
4	Optimized Fuzzy Controller Properties and Applications	46
4.1	Introduction to Fuzzy Control	46
4.1.1	Structure of a Fuzzy Controller	47
4.1.2	Operation of a Fuzzy Controller	47
4.1.3	Introduction to Defuzzification	48
4.1.4	Classical Defuzzification Methods	49
4.2	Theoretical Background	50
4.3	Applied Fuzzy Arithmetics	52
4.3.1	Arithmetics based on Inverse Functions	55
4.4	Arithmetic Based Defuzzification Method	60
4.5	Advantages of the Arithmetic Based Method	65
4.5.1	Range independence	66
4.5.2	Impact of the fuzziness	68
4.5.3	Efficiency in Calculation	69
4.5.4	Quality of the New Method	70
4.5.5	Computational Speed	71
4.5.6	Optimizing the Fuzzy Structure	73
4.6	Benefits in the Revised Methods	75

CONTENTS

Summary	87
Összefoglaló	90

Chapter 1

Introduction

There are many attempts trying to define what the word 'robot' means. Most of these definitions describe some kind of autonomous or computer controlled machine with a capability to perform some specific tasks.

However, the first appearance was introduced into the public consciousness with Karel Čapek's drama, R.U.R. (Rossum's Universal Robots) in the year 1921, where the Czech word 'robota' was referred to a synthetic humanoid instead of an electromechanical device [1].

By the second decade of the 21st century, the global robotic market became a multi billion dollar industry, where the sold units are in the range of several millions per year and growing rapidly [2]. In 2013, the total estimated market value of robot systems was nearly 30 billion dollar, while almost one third of the total value was due to the industrial robotics fueled mainly by the automotive and electronic industries [3].

To have a deeper perspective in the different designs, robotic devices should be first categorized into smaller groups. Classification can be done in various ways, for example the degrees of freedom, or the implemented level of intelligence [4], but a more useful criteria could be the classification by application or by the way of motion [5]. Industrial, domestic/household or military robots are examples for the classification by application, while wheeled, legged, flying or stationary robots are categorized by the types of locomotion and kinematics. A newly emerged field is the so-called collaborative robotics, where the human safety approach implemented in the domestic/household robots are introduced to the

operation of the industrial manipulators.

Therefore, basic researches should turn towards the application of pneumatic artificial muscles (PAMs) since their inherent elasticity makes these relatively new type of actuators optimal for the development of either collaborative or legged robots. The different aspects of PAM technology is discussed in details in Chapter 2. Another important aspect in the field of robotics is the precise trajectory tracking which is equally important for mobile and legged robots as well. The new developments on this field can be found in Chapter 3. Finally, application of a Fuzzy control system might be a good choice to develop an efficient robotic device or to operate the actuators effectively and it could be also useful for higher control levels. The details about the Fuzzy systems is discussed in Chapter 4.

1.1 Summary of the Author's Contributions

The author's contribution to this thesis is summarized in table (1.1).

Table 1.1: Results and the corresponding articles

	[6]	[7]	[8]	[9]	[10]
I.	•	•			
II.			•		
III.				•	•

Chapter 2

Pneumatic Artificial Muscle Actuator

2.1 Introduction

In the last decade, automation processes have been gaining more and more attention as the available actuators are getting cheaper and stronger. While efficiency in robotic manipulator technology has been constantly evolving, a new criteria has recently emerged, namely human safety.

Robotic devices used in the industry are usually designed with rigid joints for high speed and accurate positioning of the end effector. However, the implementation of elasticity in robotics joints and links has become mandatory to match the growing requirements of human-robot interactions [29]. Most industry leader robotic companies have already developed their human collaborative manipulator technologies, namely YuMi from ABB, LWR from KUKA, CR-35iA from Fanuc [30–32], while companies founded more recently, that address a cheaper product segment, also have come with their solutions, like Baxter from Rethink Robotics or UR3 from Universal Robots [33, 34].

All of these advanced robotic manipulators utilize some form of elasticity either virtualized with fast sensory feedback or with coupled and inherent joint stiffness. These novel techniques make a natural walking behavior also possible in specialized research robots such as ICub, Petman or Big Dog [35, 36].

Pneumatic artificial muscle (PAM) is a relatively new type of unconventional actuator

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

which can also be built into robotic joints. There are many advantages of using PAM actuators instead of a conventional one. Mostly, the extreme power-to-weight or power-to-size parameter is the main factor in practical applications. Furthermore, PAM actuators can be also utilized in robotic joints, since PAMs are inherently elastic, therefore the manufactured joint will also be flexible. Three different types of PAMs can be distinguished based on their structure components:

- Pleated muscle [11, 12].
- Straight fiber reinforced muscle [13].
- Braided fiber reinforced muscle [14–19].

Detailed description about the classification was reviewed in paper [20]. The PAM actuator used in this study is a braided fiber reinforced muscle, that means it consists of a helical braid of rigid fibers surrounding an elastic tubular bladder as seen in Fig. (2.1).

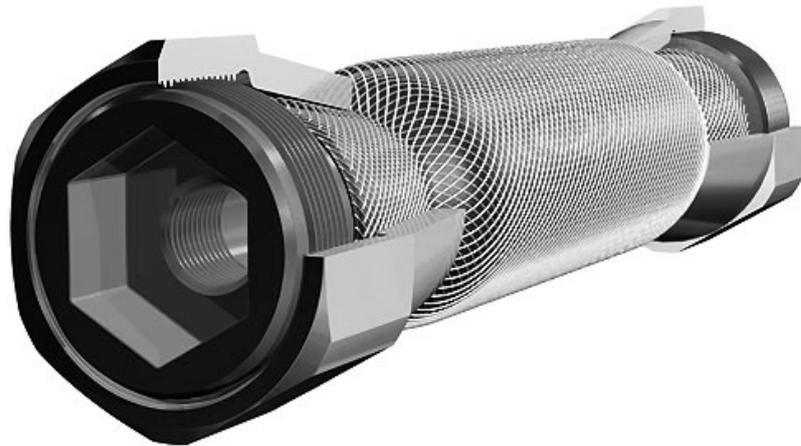


Figure 2.1: The inner structure of the FESTO fluidic muscle [21]

Upon operation, the tube is inflated with pressurized gas (usually air) or liquid material like water, therefore the braid diameter increases and the length of the actuator decreases causing a strong pulling force between the joints.

2.2 Problem Statement

Although the PAMs are very simple thus cheap constructions, controlling them is a difficult problem due to the highly nonlinear characteristics of the generated force. The main problem is that the maximum force produced by a PAM at a fixed pressure is the nonlinear function of the contraction. Basically, near 0 % contraction, also known as the relaxed state, the PAM can produce tremendous force, while this will decrease drastically and nonlinearly near the maximum contraction.

Therefore, accurate modeling of the actuator is necessary to develop a reliable and precise system when using a PAM component. There are two main approaches which can model the behavior of the PAM as an actuator. One of these focuses on the static characteristic of the actuator, where the measurements are done in equilibrium states, while other models are about to describe the dynamic properties of the PAM actuator.

In the following section, the former approach will be presented using only the static behavior of the PAM with a transfer function which describes the correlation between the muscle contraction, the relative pressure in the muscle and the generated force.

Another main advantage using PAMs as actuators is that these can be built into robotic joints without applying heavy and complex gearing mechanisms such as epicyclic drive [37] or harmonic drive [38]. However, the nonlinear characteristics of PAMs make these actuators difficult to use in these applications as well. As a result, the nonlinear force generated by a PAM actuator will be converted into a rotational movement with similar nonlinear torque characteristic when using simple circular gearing mechanism [39], or mounted in a curved fashion [40]. The basic idea presented in the second part of this Chapter was to construct a spiral and cable transmission unit with a non-circular cable spool in order to achieve better force to torque converting capability.

2.3 The Transfer Function

An exponential function was first selected as a core function to describe the static correlation between the force (y), contraction (κ) and the required pressure (p). The exponential function provides the best approximation for the highly dynamic change of the measured data points instead of using trigonometrical or low ordered polynomial.

The following criteria should be satisfied:

- The function must fit any isobaric measurement series in the permissible operating range (see the FESTO catalog [22]).
- The number of the unknown parameters should be as few as possible.

The exponential function in equation (2.1) uses only four unknown parameters and it can express the correlation between contraction and force for a fixed pressure.

$$F_{core}(\kappa) = a \exp\left(\frac{1}{\kappa+b}\right) + c\kappa + d \quad (2.1)$$

Where (a, b, c, d) are unknown parameters.

The next step is to make some parameters of the core function pressure dependent resulting in a general function called transfer function. One of the main benefits in using a simple function instead of an algorithm is the ability to invert it. The inverse form of the transfer function can be used as an open-looped positioning system [23]. For this step the following criteria should be satisfied:

- The transfer function must fit the whole permissible operating range.
- In order to make the parameters pressure dependent, only a limited number of new parameters should be added to the core function to preserve simplicity.
- The general function should be invertible to express the required pressure for a desired contraction and force.

The general form of the core function in equation (2.2) is using only six unknown parameters.

$$F_y(p, \kappa) = (ap + b) \exp\left(\frac{1}{\kappa+c}\right) + (d\kappa + e)p + f \quad (2.2)$$

Where (a, b, c, d, e, f) are unknown parameters.

From equation (2.2) the variable p can be expressed by equation (2.3) as an inverse function.

$$F_p(\kappa, y) = -\frac{b \exp\left(\frac{1}{\kappa+c}\right) + f - y}{a \exp\left(\frac{1}{\kappa+c}\right) + d\kappa + e} \quad (2.3)$$

Where (a, b, c, d, e, f) are the same parameters from equation (2.2).

Note that equation (2.2) can be given as a more realistic formula expressing the permissible operating range. For instance, in equation (2.4) the force limiting property was added.

$$F(p, \kappa) = \begin{cases} F_y(p, \kappa), & \text{if } 0 \leq F_y(p, \kappa) \leq y_{max} \\ y_{max}, & \text{if } y_{max} < F_y(p, \kappa) \\ 0, & \text{if } 0 > F_y(p, \kappa) \end{cases} \quad (2.4)$$

Where y_{max} is the highest force in the operating range, or the force compensation limit in the force limited muscle. The limits for κ can be added similarly with trivial modification of equation (2.2).

2.4 Experimental Setup

All measurements were conducted on an experimental setup capable of measuring the pressure, contraction and the force generated by the PAM actuator at the same time. Main components of the test-bed are the followings:

- Motorola MPX5999D pressure sensor
- LINIMIK MSA 320 type linear incremental encoder

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

- MOM 7923 type strain gauge force sensor
- FESTO VPPM-6L-L-1-G18-0L6H-V1N-S1-C1 proportional valve
- Lab PC 1200 National Instruments multi-IO card

The assembled test-bed is depicted in Fig. (2.2).

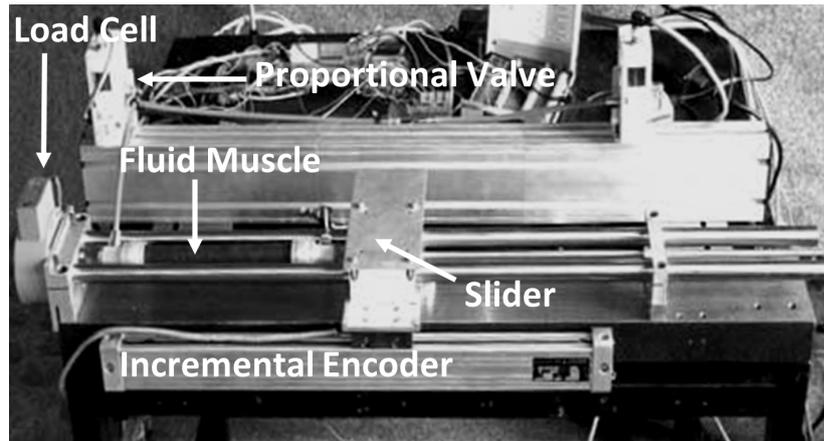


Figure 2.2: Photo of the test setup [24]

The working principle of the setup is simple. One end of the PAM actuator is attached to the force sensor (left side on the photo), while the other end is linked to a movable cart with an adjustable length rod (middle and right side on the photo). The movable cart is also attached to the linear encoder (bottom on the photo) and all other main components including the valves are in the background. The encoder can only measure the distance relative to a programmable reference point. All measurements have to be done with the same initial reference point to achieve valid and accurate data series.

Also note, to make the transfer function valid in general, the contraction of the PAM has to be considered as a relative contraction instead of the absolute change of displacement. This can be calculated as the signed change of displacement in percentage relative to the total length, where a negative value means that the actuator is expanded further than the resting condition. The inner pressure of the actuator can be adjusted with the proportional valve, while the inner pressure and force can be recorded together with the calculated relative contraction. For detailed description of the setup see reference [24].

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

For experimental validation, the FESTO DMSP-20-400 PAM was used with the test-bed to obtain the reference data series which consist of 340 measurements. Thirty measurements were recorded for each fixed pressure value from 0 to 5.5 *bar* pressure with 0.5 *bar* steps in the working range. Each isobaric data series were equidistantly recorded starting from the relaxed contraction state up to -3% contraction. Finally, data points were removed due to permissible operating range limits meaning that only smaller than 1500 *N* forces and smaller than 25% contractions were used as suggested by the FESTO catalog [22].

2.4.1 Model Prediction Capabilities

The transfer function published in 2009 first was used for comparison, see reference [25] for more details. This study suggested to use function (2.5) with eight parameters as a transfer function.

$$F_{prev}(p, \kappa) = (ap + b) \exp^{(c\kappa + d)} + (ep + f) \kappa + gp + h \quad (2.5)$$

Where (a, b, c, d, e, f, g, h) are unknown parameters.

To compare the prediction capability of function (2.5) with function (2.2), identical circumstances were created for the search of the unknown parameters. The goal of this search was to find those parameters that give the best estimation of the cleaned dataset described above. For finding those parameters genetic search was used with the following properties:

- Fitness function: root-mean-square error (RMSE) of the predicted values for the cleaned dataset.
- Population size: 40.
- Generations: 10000.

Table (2.1) shows the best RMSE values after the genetical algorithm search.

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

Table 2.1: Best RMSE values for the predicted forces

	$F_{prev}(p, \kappa)$ function (2.5)	$F(p, \kappa)$ function (2.2)
RMSE value [N]	6.93N	6.44N

The presented function gives approximately 7.1 % smaller RMSE value. Also note, the previous function is more likely to fall into the local minimum during the search due to the higher number of parameters or also known as the curse of dimensionality, while the presented function tends to be more stable. The prediction capability of the presented function can be seen in Fig. (2.3) with the whole dataset. The data points are depicted for the working range from 0 to 5.5 bar pressure with 0.5 bar step, where the lowest series represents the 0 bar and the highest series the 5.5 bar measurements.

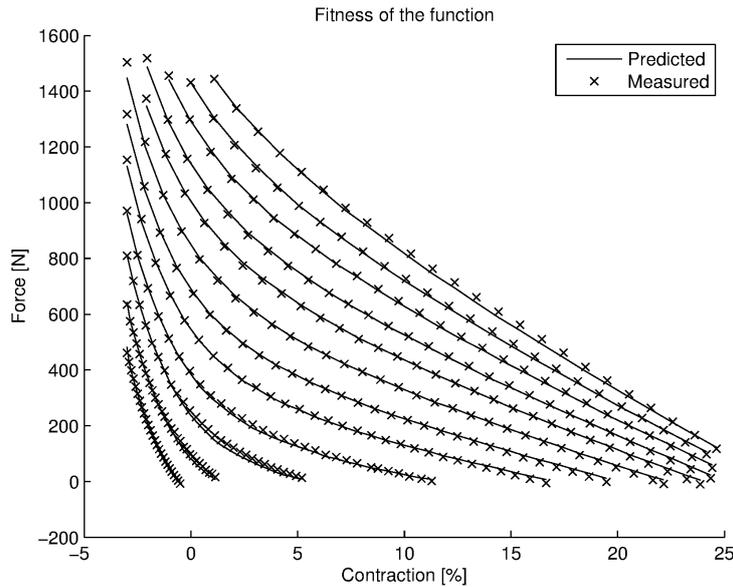


Figure 2.3: Comparison of measured and predicted data

Furthermore, the error function was also obtained from the difference of the measured and predicted forces to compare the prediction capability of the presented model (2.2) with the previously described model (2.5). The error functions, depicted in Fig. (2.4) and Fig. (2.5), consist of three data series for fixed 0, 2.5 and 5.5 bar pressure, respectively.

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

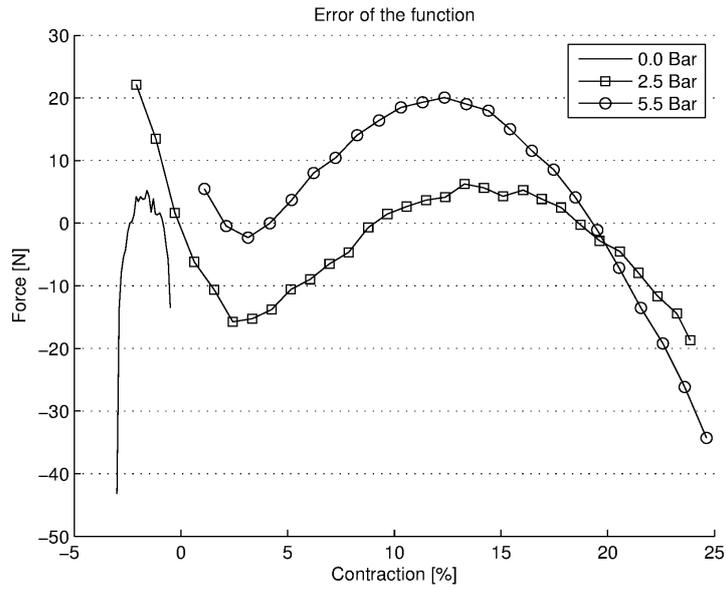


Figure 2.4: Error function for the previous model (2.5)

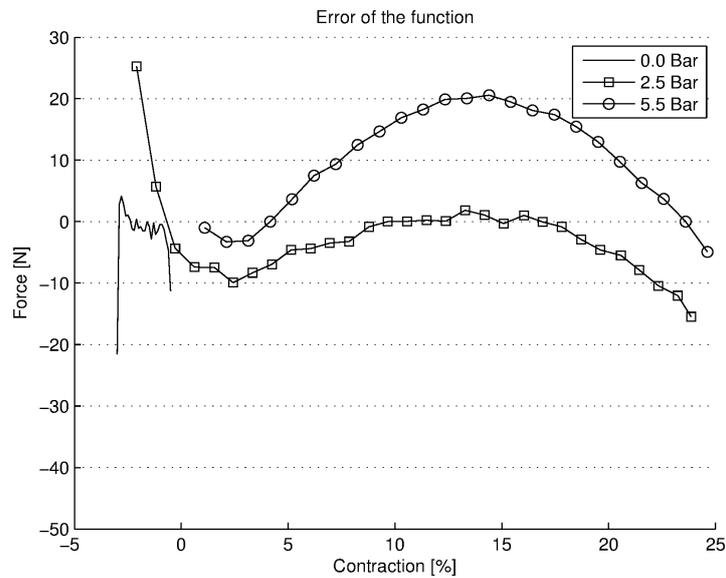


Figure 2.5: Error function for the presented model (2.2)

If the theoretical optimum for the error function is constant zero for all cases, the model fits perfectly for all the measured data points. As seen in Fig. (2.4), the error of the previous model is significantly higher at low pressure (0 bar) and high contraction ($> 20\%$) compared to the presented model. Furthermore, the error function of the presented model tends to be

near zero for many cases especially at the usual working range (2.5 bar) as shown in Fig. (2.5).

2.4.2 Calibration Procedure

The basic component of the FESTO PAM actuator is the inflatable elastic rubber tube. In a typical application using this type of actuator, the inner pressure continuously changes, thus the elastic material will become harder and the performance will deteriorate. To compensate this behavior of the PAM actuator, regular calibration is required after installation of the system. As a result, the parameters must be regularly readjusted when using the proposed function as part of a controller.

The calibration of the system can be quite time consuming, since several hundreds of different states must be measured. To lower the required number of measurements, three different pressure values with three different contraction ratios were chosen as a total of nine representative data points for the whole operating range. These points were at low (0 bar), medium (2.5 bar) and high (5.5 bar) pressure with the lowest, medium and the highest contractions, respectively as seen in Fig. (2.6).

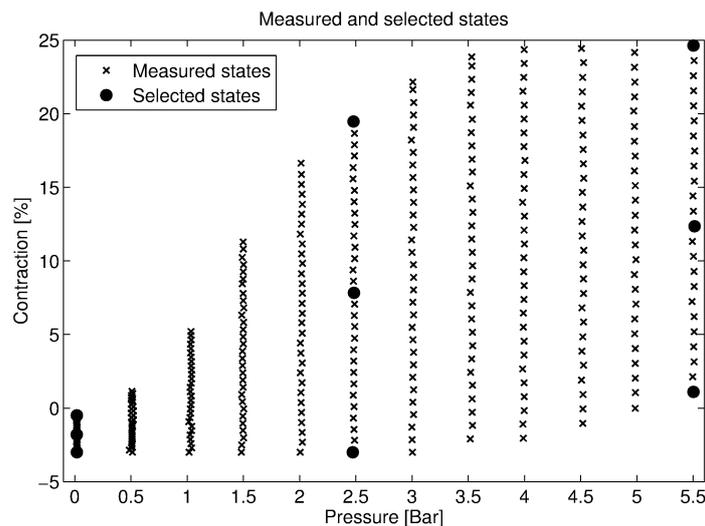


Figure 2.6: For calibration, nine states were selected from the whole operating range.

As a result, the following Fig. (2.7) shows the function (2.2) which was fitted on the

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

above mentioned nine selected states. It is remarkable to note, that the resulting function is more precise at the extreme working range when only nine states were used for the calibration. However, the average fitness is slightly less accurate compared to the particular case when all 340 states was applied to fit the model.

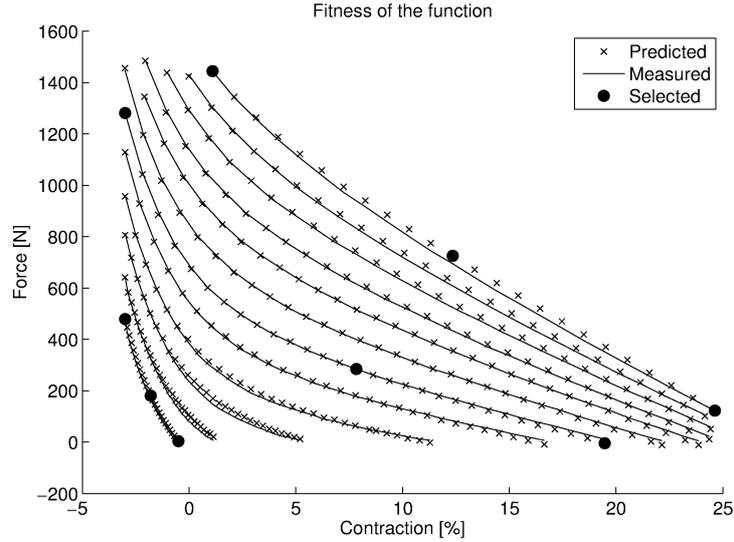


Figure 2.7: Comparison of all the measured data with the predicted based on the selected states

Next, the prediction capability of function (2.5) was compared with function (2.2). For this purpose, identical circumstances were created for the search of the unknown parameters which involved the fitness function, the size of the population and number of generations as detailed in the previous subsection. Table (2.2) shows the best RMSE values after the genetical algorithm search.

Table 2.2: RMSE values for the predicted forces trained with the selected data points

	$F_{prev}(p, \kappa)$ function (2.5)	$F(p, \kappa)$ function (2.2)
RMSE value [N]	11.53N	9.37N

After training on nine selected and evaluated to the total of 340 measured data points, 9.37 N RMSE was achieved with function (2.2), while function (2.5) gives 11.53 N RMSE

which is approximately 18.7 % worse compared to the predicting capability of function (2.2).

2.5 Benefits of Using the Developed Transfer Function

The transfer function of PAM actuators describes the correlation between the muscle contraction, the relative pressure in the muscle and the generated force. The precision and simplicity are the main aspects that must be considered when creating these functions. Numerical studies aimed to describe simple transfer functions where the number of the unknown parameters are relatively low (< 10) [25–27]. For instance, Kerscher et al. [27] introduced a function with only five unknown parameters, but the model error is higher at lower pressure range. In the contrary, Hosovsky et al. [28] showed a highly accurate model, although as many as 189 unknown parameters were required to be determined. In 2009 [25], a relatively compact yet accurate model was presented using eight parameters capable of modeling the static characteristics of PAM actuators. The presented function further improved the compactness and precision, therefore a new simple force function was developed and compared with the model introduced in reference [25].

The benefit in using this function is the lower error of it, especially in the usual working range. It uses only six unknown parameters resulting in faster and reliable parameter search. Furthermore, its inverse function is capable to express the pressure required to maintain a given contraction with a given force. This property of the function is necessary in an open-looped controlled system or generally, when it is not possible to take direct measurements for all three parameters but one of them can be expressed from the others. The compactness of the function results in high evaluation speed necessary for fast controllers and for modeling complex systems equipped with PAM actuator. In case of recalibration, only a limited number of measurements are required to refit the model with remarkable precision. This capability can be used to develop real-time calibration system that has longer operating time in between regular maintenances.

2.6 Using PAMs as actuators

In order to build a useful PAM actuator, the nonlinear behavior should be balanced at the edges of the operating range. The conversion of the forces into a rotational torque generated by an antagonistic pair of PAM actuators is depicted in Fig. (2.8).

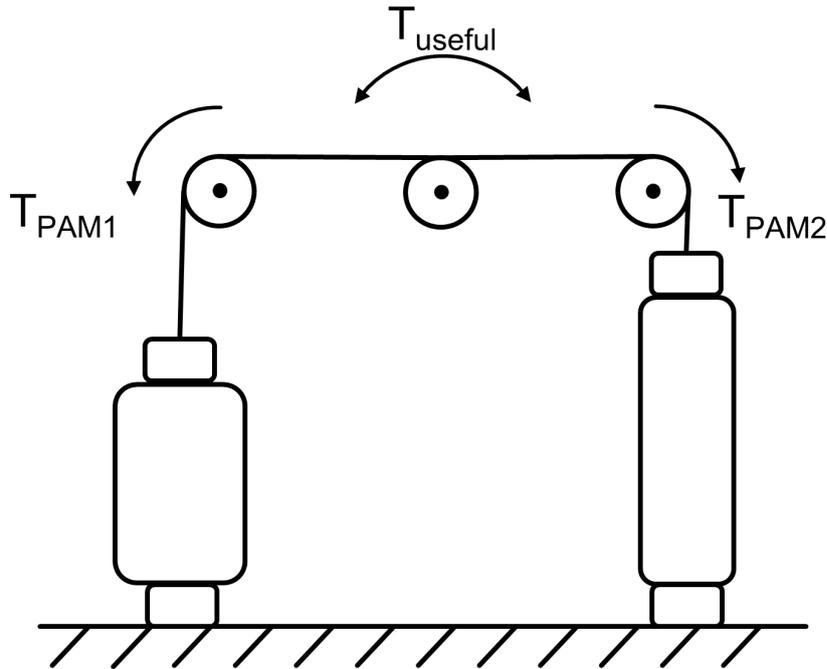


Figure 2.8: The basic antagonistic setup of PAM actuators with standard disc shaped gear system

In this case, PAM1 and PAM2 produce antagonistic forces which are transferred into torsional torque via the center disk, where the resulting T_{useful} torque can be expressed with the following equation (2.6)

$$T_{useful} = \min(T_{PAM1}, T_{PAM2}) \quad (2.6)$$

T_{useful} torque is also depicted in Fig. (2.9), with the T_{PAM1} and T_{PAM2} , which are the exerted torques generated by the PAM actuators on the outer disks.

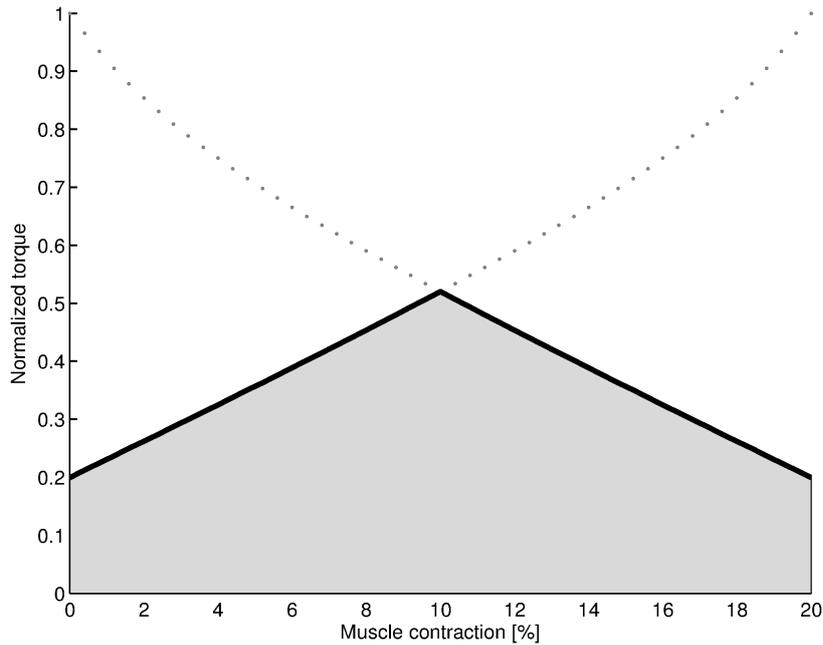


Figure 2.9: The normalized torque values (dotted lines) in an antagonistic setup of PAM actuators using simple circular gearing mechanisms. T_{useful} torque range is shown by the grey area as a function of muscle contraction at the maximum operating pressure. The Y axis is normalized in order to show scale independent torque values

The Fig. (2.9) clearly shows that near the edges of the operating range, the T_{useful} torque is less than one third and can barely reach higher than half of the theoretical maximum at its peak.

To solve these problems, it is necessary to level off and increase the useful torque as much as possible at the whole operating range. This can be achieved by applying a non-circular gearing mechanism, which is capable of reducing the torque generated by the PAM near the minimum, while increasing it near the maximum contraction.

The aim of the present study was to develop a framework, which is capable to simulate similar transmission mechanisms as seen in Fig. (2.10). The framework should support various ways of defining the disk geometry, which are not limited by closed form functions as seen in reference [41], but can also be expressed numerically. The non-circular shaped geometry should also be optimized by certain profile or criteria to enhance the effectiveness

of the transmission. To enhance the utilization of the torque generated by PAMs, a specific non-circular disk geometry was optimized with the framework. An experimental device was also constructed to validate the simulations, which allow to test different disk shapes, and measure the correlation between the contraction of the simulated PAM and the rotation of the central disk.

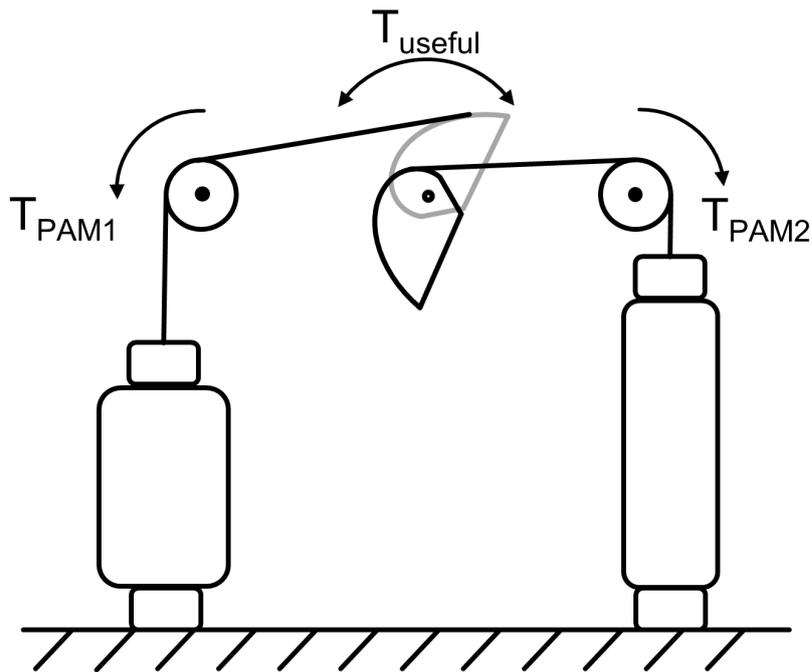


Figure 2.10: The setup using two non-circular cable spool element with antagonistic configuration of PAM actuators

2.7 Designing Principals of the Simulator

The non-circular element will take a crucial role in the transmission as seen in Fig. (2.10) and its connection to the PAM actuator must be carefully designed to develop an effective robotic manipulator. To support various sized and shaped pulleys, an intermittent disk should be applied between the PAM and the pulley as seen in Fig. (2.10). The role of this disk is to direct the linear motion of the cable from the PAM towards the non-circular element.

The mathematical approach used for handling the pulley geometry is also an important

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

designing principal. In this study, the pulley geometry must be given by finite amount of control points, thus all the tangents and the convex hull are calculated easily with simple algorithms. Using sufficiently dense discrete control points will result in a smooth physical surface of the disk during manufacturing, because most commercial CNC machines has finite resolution in the range of 0.01 mm . These control points can be given empirically or can be derived from a parametric function, or even from the combination of these methods.

The following criteria were used for the simulations:

1. The stiffness of the cable is infinite

With modern braided type fishing line products made from non-stretching materials, a wide range of durable cable material can be obtained at for this purpose.

2. The cable diameter is negligible

The available cable materials, like braided Kevlar line is advanced enough to have near one thousand Newton tensile strength with around one millimeter diameter, which is negligible. Also note if higher precision or higher scaling up of the whole system is required, it is easy to compensate all the calculations with the radius of the cable.

3. The radius of the intermittent disk is not negligible

It would be unpractical to make the intermittent disk with relative small radius because of the following reasons. The cable friction at the rotation point would become significant, causing decreased efficiency and increased wear out of the transmission.

4. The α orientation of the pulley is defined such that it satisfies equation (2.7)

$$\alpha_{min} \leq \alpha \leq \alpha_{max}; (\alpha_{max} - \alpha_{min}) \ll 360^\circ \quad (2.7)$$

Like the range of all human limb joints, most robotic manipulator joints have usually less than one complete revolution, therefore this constraint is not an actual limitation for many robotic applications. Also note, if wider rotation range is required, it is

possible to serially connect two transmissions actuated by two pairs of PAMs from both sides of the linkage.

2.8 Implementation of the Numerical Simulation Constraints

The designing principles described in the previous section give enough constraints to obtain all the equations necessary building the framework. The essential part of the transmission with the three main sections of the cable is depicted in Fig. (2.11A).

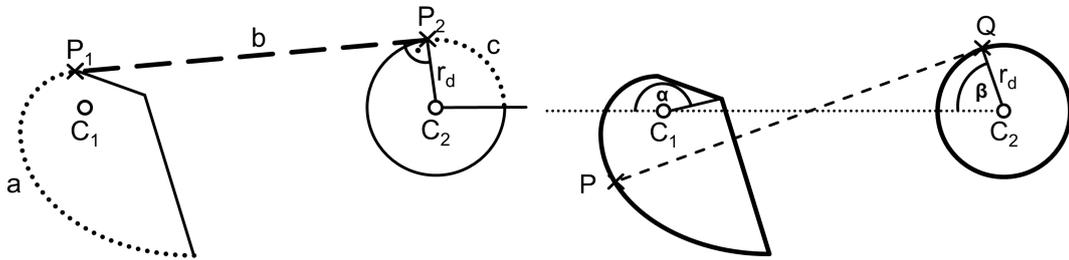


Figure 2.11: (A) The overall layout of the cable sections, while (B) shows how to search the P point that will give the biggest β for any given α . If more than one P points give maximum β value, the one with the shortest PQ length is chosen

The framework should be capable to calculate also possible for any given α :

1. The convex hull of the non-circular disk geometry

If the PAM is contracted, the cable will be stretched, thus the cable cannot follow any non-convex shape by its nature. Although, the calculation of P_1 and P_2 points will give the same result for both the original geometry and its convex hull shape, in order to calculate the spooled a part of the cable, it is recommended to use the convex hull geometry. Also note, there is no practical use in designing non-convex spool for this type of transmission therefore, the convex hull conversion is only a non-critical security feature of the framework.

2. The P_1 and P_2 points of the mutual tangential line b

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

Finding the valid b mutual tangent for a given α orientation of the non-circular disk is equivalent with finding the tangent from all P points to the upper semicircle of the intermittent disk with the biggest β value as seen in Fig. (2.11B). Note, it is true only if the biggest radius of the non-circular disk is smaller than the $distance(C_1, C_2) - r_d$. Furthermore, it is possible to use an exhaustive search in order to find the mutual tangential line b because in practice, there are only a few thousand P control points defined on the non-circular disk.

To proceed with the calculations, the points should be given with reference to a 2D coordinate system, as follows:

$$C_1(0,0); C_2(C_2X,0); P(PX,PY); Q(QX,QY) \quad (2.8)$$

Thus, for any given α orientation that satisfies (2.8), the following equation (2.9) gives β

$$\beta = \sin^{-1}(PY/distance(P,C_2)) + \cos^{-1}(r_d/distance(P,C_2)) \quad (2.9)$$

The missing QX and QY values can be expressed with the following equations (2.10)

$$QX = C_2X - (r_d \cos(\beta)); QY = r_d \sin(\beta) \quad (2.10)$$

While P_1 point will be equal to the P point with biggest β value, P_2 point will be equal to the Q point corresponding to the same biggest β value.

Also note, it is not always necessary to use the basic brute force search to find the mutual tangent, because in some cases, simple heuristics can significantly reduce the number of control points to be examined. As an example, usually it is enough to test only the P points with positive PY value.

3. The amount of cable wound up onto the disks

It is clear from Fig. (2.11A) that all the l cable wound up to the transmission unit can be decomposed into three subsections, as the following:

$$l = a + b + c \quad (2.11)$$

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

It is also clear, that all these subsections of the cable depend on the α orientation of the non-circular element, so as their sum will also be α dependent:

$$l_\alpha = a_\alpha + b_\alpha + c_\alpha \quad (2.12)$$

In order to get the lengths of these segments, P_1 , P_2 and must be calculated first as described previously. Summing up the consecutive control points on the convex hull of the non-circular geometry starting from point P_1 will give the actual value of a_α .

The distance of point P_1 and point P_2 will give b_α , while c_α can be calculated as follows:

$$c_\alpha = r_d(180^\circ - \beta) \quad (2.13)$$

4. The designed κ contraction of the PAM as a function of α

The basic assumption is that the PAM actuator should be in relaxed state when the non-circular disk α position is at the lowest end of the designed range such as $\alpha = \alpha_{min}$. Similarly, the designed maximal κ_{max} contraction rate should be reached in case of $\alpha = \alpha_{max}$. With all these assumptions, the following equation will give the κ_α :

$$\kappa_\alpha = \frac{\kappa_{max}(l_\alpha - l_{\alpha_{min}})}{(l_{\alpha_{max}} - l_{\alpha_{min}})} \quad (2.14)$$

5. The maximum torque of the transmission system

First, a static model of the PAM is required, which is capable of expressing the force of the PAM as a function of contraction and pressure. An accurate model can be found in [41] for this purpose. If F_{κ_α} represents the maximum force of the PAM at κ_α contraction, than the maximum T_α torque of the system for any given α can be expressed with the following:

$$T_\alpha = distance(b_\alpha, C_1)F_{\kappa_\alpha} \quad (2.15)$$

2.9 Experimental Results

The testing of the framework was carried out through a virtual upgrade of a regular disk shaped transmission system with torque amplification in mind. First, a regularly designed transmission was specified based on the layout depicted in Fig. (2.8). This system is considered as a reference case, which has a 180° rotation range, which is also common in geared servo motors. The distance between C_1 and C_2 was set to 5 cm to keep enough space for the PAM actuator. The radius of both the central and intermittent disk was set to 10 mm . The modelled actuator was a 20 mm in diameter Festo PAM operated at 5 bar pressure described in [6]. The maximal contraction was set to 20% , so the total length of the modelled PAM was $5\pi r_d \approx 157\text{ mm}$. The modelled torque characteristic of the reference transmission can be seen in Fig. (2.9), while the actual torques measured in $[Ncm]$ can be obtained by multiplying all values with $1415N$. For the entire modelled transmission the peak maximal torque would be $T_{peak} = 736.46\text{ Ncm}$, while the average torque would be $T_{mean} = 501.79\text{ Ncm}$.

The main goal of the next section is to improve the torque characteristic of the reference transmission system, while all other parameters must remain the same, including the length of the PAM, and the total rotation range of the transmission. Only the central disk element can be altered.

2.9.1 Design Optimization with Genetic Algorithm

A non-circular disk geometry should be developed for the reference design described in the previous section. The layout of the shape can be expressed as depicted in Fig. (2.12).

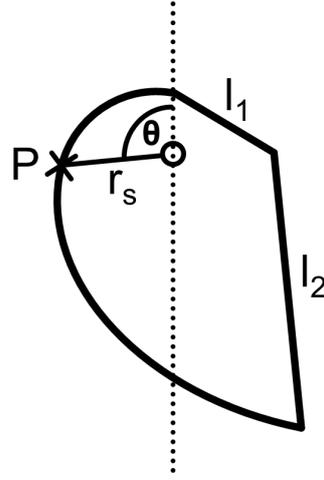


Figure 2.12: The general layout of the non-circular shaped disk, where P is given by r_s radius as a function of θ angle, while l_1 and l_2 are linear segments on the pulley

The linear l_1 and l_2 segments makes the whole geometry easier to construct, also the shorter l_1 segment was used for calibration purpose later. For this example r_s was given as a third order polynomial function of the θ angle as follows:

$$r_s(\theta) = f_1\theta^3 + f_2\theta^2 + f_3\theta + f_4 \quad (2.16)$$

The missing f_i coefficients were find with genetic algorithm, with the following specifications. The genetic representation was simply the f_i vector with the four element of the coefficients, while the fitness function was the sum of the difference of the two torque function depicted in Fig. (2.9). In order to minimize this function the algorithm must find a geometry that will weaken the PAM near minimum contraction, while makes it stronger near the maximum contraction. This is possible if the geometry has shorter and longer radiuses at the opposite sides, similar to the one depicted in Fig. (2.12). The torque characteristic of optimized geometry after 1000 iteration can be seen on top of the reference characteristic in Fig. (2.13).

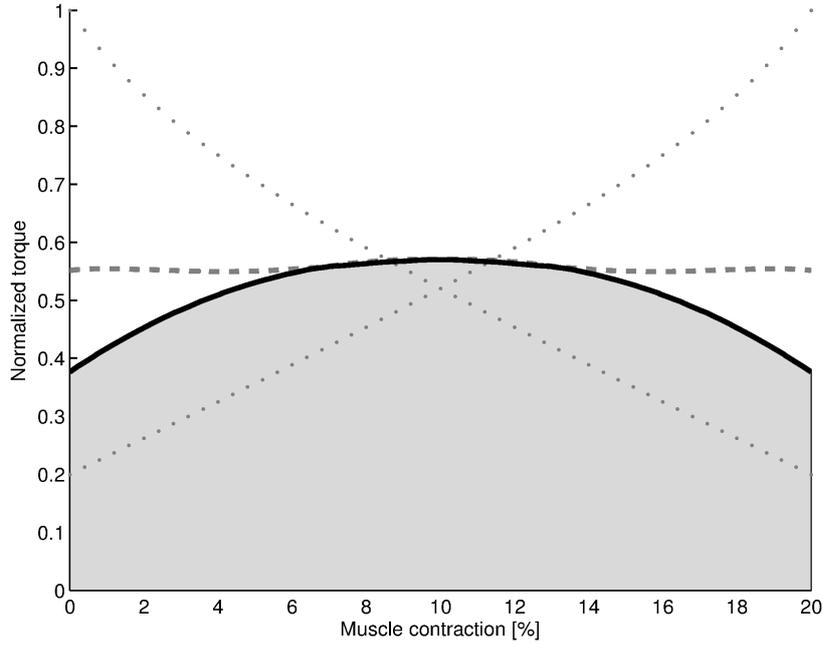


Figure 2.13: The normalized torque values (dashed line) in an antagonistic setup of PAM actuators using the optimized non-circular shaped disk element as a gear system. T_{useful} torque range is shown by the grey area as a function of muscle contraction at the maximum operating pressure, while the Y axis is normalized to the reference design torques (dotted line) in order to make them comparable

The result is a significant improvement of the whole torque range. The actual torque values can be obtained by multiplying the normalized values with 1415 N , thus the maximum torque would be increased to $T_{peak} = 806.64\text{ Ncm}$, while the average torque would be increased to $T_{mean} = 716.59\text{ Ncm}$. Also note that, near the edges of the operation range the optimized torques are almost doubled relative to the reference design, which is significant because only the central disk was replaced, but all other parameters remained the same.

2.9.2 Validation of the Design Using an Experimental Device

An experimental setup was developed to test the transmission with different geometries. The arrangement of the device is simple therefore, it is easy to scale up or customize. Fig. (2.14) shows the assembled test setup.

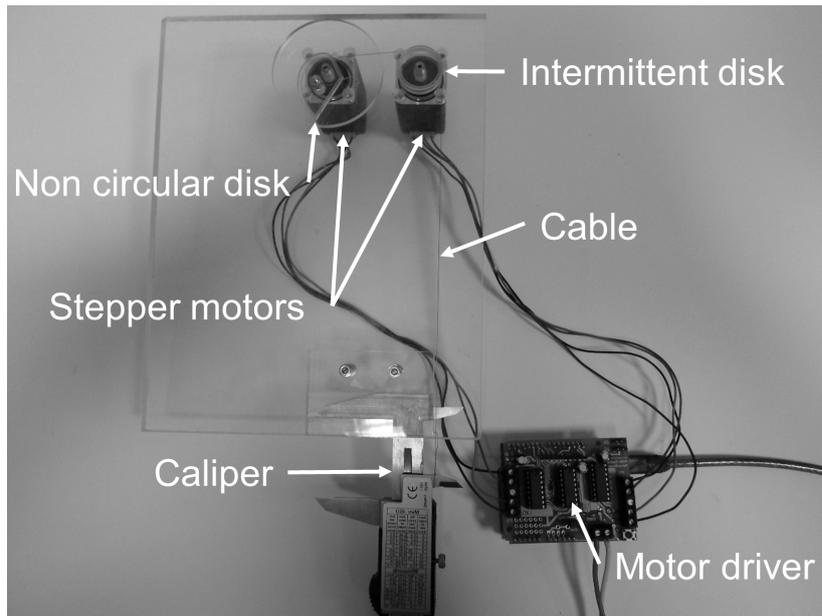


Figure 2.14: The image of the experimental setup. The different parts of the device are labelled on the photo

As seen in Fig. (2.14), there are a few major components of the device. The electrical board is an Arduino One with a motor shield on the top of it. Both the non-circular and the intermittent disks can be driven separately with two stepper motors, which are also used as bearings. The movement of the cable can be measured by a digital caliper with a resolution of 0.01 mm . As a cable material for the model, a braided catfish leader was used, which is a relatively common type of fishing gear. For the validation tests, the Sufix Gyro 0.14 mm diameter braided line was used, which is a very soft, but extremely stiff cable with a breaking strength of 53 N . In a real transmission, this cable can be easily scaled up to the required 1415 N with a cable still under 2 mm in diameter. The whole framework and all the disks were precision milled from plexiglass with a CNC machine at an accuracy of 0.02 mm . In order to make the G-code, a special convex-hull algorithm based script was designed with a capability to compensate the tool diameter.

The measurement was started with a calibration, when the caliper was set to zero with a well defined θ angle of the non-circular element. After the calibration, the stepper motor was commanded to rotate in 5 step bursts to wind up the cable, while the position of the caliper

was recorded after each sequence. The same process was simulated by the framework and the experimental results of the transmission model are compared in Fig. (2.15).

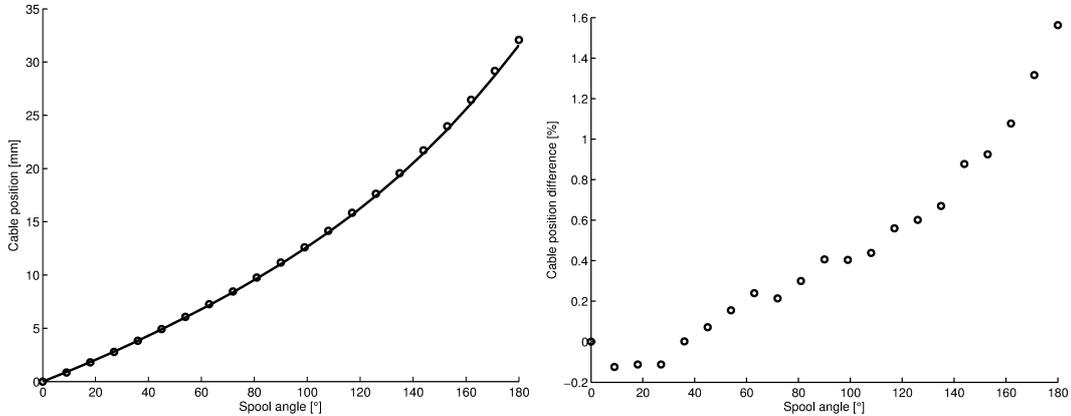


Figure 2.15: (A) Shows the measured (circles) and the simulated cable positions (continuous line) at different θ spool angles. The difference between the two series is presented in percentages in (B)

The results showed that the measured and the simulated values are so similar, that the difference is noticeable only in Fig. (2.15B), where most of the error was under 1 %. Also note, the error diagram showed a similar tendency to the original measurements, which was most likely derived from a slight miscalibration. The purpose of the fabricated model was to validate the correlation between cable movements and the corresponding rotation of the spool instead of measuring the torques generated by a given force on the cable. However, in a real transmission unit made of stronger materials, the torque characteristic could be verified, and the error could be further eliminated with a more precise calibration protocol.

2.10 Benefits of Using the Spool and Cable Transmission System

In this work, a simulation framework was developed which is capable to optimize nonlinear transmission systems with various characteristics. In the heart of the transmission, a non-circular pulley converts the motion of a cable into a rotating torque. The correlation between

CHAPTER 2. PNEUMATIC ARTIFICIAL MUSCLE ACTUATOR

the orientation and the radius of the central disk is defined with a finite vector pair, therefore it can be expressed with either a closed form function or any other empirical way. The framework can simulate the momentary gear reduction of the transmission as a function of the orientation of the central disk. Also, by continuously tracking the cable alignment and the cable length spooled onto the central and intermittent disk elements, the framework is capable to express the gear reductions as a function of cable position. These functions makes the simulation environment suitable for optimizing transmissions for PAM actuators.

An experimental transmission unit was designed for PAM actuators with a common, 180° rotation capability using the framework, where the aim was to increase the expressed torque for the whole operating range. Comparing this optimized system with a regular transmission that utilize only a disk as a central rotating element, the increase of the average torque was 42.8 %, the increase of the peak torque was 9.53 %, while the range of the unit and all of other components remained the same.

As a validation, the simulated transmission was fabricated using the optimized non-circular disk geometry, to test the gear reduction capabilities of the physical model. The results shows high correlation with the simulated model behavior, where the difference between the simulated and measured values of the central disk orientation and the cable positions were under 1.6 %.

Chapter 3

Pure Pursuit Trajectory Tracking

3.1 Introduction

Mobile robots attract great attention due to the wide field of applications which ranging from industrial over services and military to consumer products applications. Such robots can be classified generally into wheeled mobile robots (WMRs) and legged ones, in which the former can have a large number of possible wheel configurations and kinematic designs. Each configuration has its merits and demerits with respect to a certain application. The WMRs can be classified into five configurations according to the degree of mobility and the degree of steerability; the degree of mobility is defined as the number of WMR velocities that can be assigned instantaneously and independently, while the degree of steerability is specified as the number of orientable wheels that are steered independently [42, 43]. The omnidirectional, the differential-drive, the syncro-drive and the car-like mobile robots are some configurations of WMRs. Among those configurations, the differential-drive configuration is the most common robot configuration that can be used because of its simplicity and versatility.

Navigation is one of the most important research areas for WMRs. Trajectory planning and tracking are common autonomous motion control problems for navigation in known and unknown environments. Trajectory planning is a task of generating appropriate path, which is an explicit function of time to reach a particular location while avoiding collision

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

with obstacles. Trajectory tracking is a task of autonomously driving the robot along the planned path by continuously commanding the robot with calculated speeds that compensate the tracking errors.

Planning a smooth trajectory that is twice differentiable function of the time is important for the smooth motion of the robot. Trajectory planning can be classified generally into local, global, and optimal planning. Local planning depends on immediate sensor data while global one uses a prior knowledge of the environment. The potential fields [44], the vector field histogram [45] and the dynamic window [46] are three common approaches for local planning. Global planning can be categorized into cell decomposition [47] and graph based (road-map) [48] approaches. Optimal planning is concerned with planning a trajectory that is optimal according to a certain cost function to satisfy a criterion or a constraint. The minimum time to reach a goal and the shortest distance are some examples for a criterion to be satisfied, while avoiding the violation of the maximum velocity and acceleration are those for a constraint. The use of smooth parameterized splines that pass through waypoints, specially the cubic spline is an effective method to describe this optimal trajectory [49, 50].

To execute the planned trajectory not only trajectory tracking but also posture stabilization is required. Posture stabilization is to stabilize the position and orientation of the mobile robot at the same time. Since, nonholonomic systems cannot be asymptotically stabilized around an equilibrium using smooth time-invariant feedback [51], the trajectory tracking is much easier to accomplish than posture stabilization. Posture stabilization is accomplished by smooth time-varying feedback [52] or discontinuous controllers [53]. Some of the controllers which achieve both the trajectory tracking and the posture stabilization simultaneously were proposed in [54, 55], in which the stabilization problem was converted to an equivalent tracking problem. The well-known simple and effective state-tracking controller which based on Lyapunov stability was proposed in [47, 56], in which the system's equations are linearized with respect to the reference trajectory. In this method, the control strategy combines the feedforward solution and the feedback action. Dynamic feedback linearization approach was also presented in [55, 58] for the problem. Predictive control techniques are

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

proposed for the trajectory tracking in which the linear model of robot kinematics is used to predict the future system outputs [59].

The geometric-based pure pursuit method was first introduced in the field of mobile robotics in [60] for the path tracking. It is based on commanding a robot to follow a curve that connects the actual position of the robot and a goal position on the reference path which is fixed distance ahead. Circular curves were used in [61] to connect those two positions, while quintic polynomials were used in [62] for this purpose. The stability of this method was studied in [63]. The lookahead distance parameter developed in the original pure pursuit method is a fixed distance. It is further developed to be determined using a real-time fuzzy logic system that based on the path characteristics, velocity and tracking errors [61]. Obviously, the original pure pursuit method is a pure path tracking method which does not include the time required to accomplish this path. So, in order to make it a feasible and reliable trajectory tracking method that can consider the execution time and can cope with the velocity and acceleration constraints imposed by the robot, another approach is needed which is the contribution of this work. In the developed approach here, the fixed lookahead distance is radically changed to be an updating time dependant lookahead distance considering the planned trajectory which is implicitly satisfying those constraints. Namely, the developed approach calculates a lookahead distance between the current position of the robot and the desired position of the robot on the planned trajectory after n time steps, which can be determined based on achieving both the robot stability and high accuracy tracking errors. From this time dependent lookahead distance, the curvature of a circular arc connecting those two positions can be determined and thus the commanded linear speeds of the wheels can be calculated in real-time. Therefore, the developed approach makes the original one a real-time trajectory tracking approach that can achieve high accuracy tracking errors with high reliability, and thus it can compete with the commonly used state tracking controllers. Moreover, the developed method avoids the selection of the tuning and damping parameters used by those well-known state tracking controllers.

3.2 Optimal Trajectory Planning Algorithm

The objective of the optimal trajectory planner is to plan a trajectory that is connecting the initial and final positions and velocities in the shortest time and satisfying certain dynamic constraints. Avoiding the violation of the maximum velocity and acceleration imposed by the robot is an important constraint to be satisfied to make the dynamic effects negligible. In this study, a trajectory planner is implemented based on a previous work described in [50]. Briefly, the parameterized cubic spline which is twice differentiable function of the time is used to describe the required optimal trajectory interpolating waypoints. The x and y components of the trajectory segment points as a function of the time t are given as follows:

$$x_d(t) = b_x t^3 + c_x t^2 + d_x t + e_x \quad (3.1)$$

$$y_d(t) = b_y t^3 + c_y t^2 + d_y t + e_y \quad (3.2)$$

where the coefficients $b_x, b_y, c_x, c_y, d_x, d_y, e_x,$ and $e_y,$ can be determined using the initial and the final positions and velocities of the segment; the maximum velocity, $v_{max},$ and the maximum acceleration, $a_{max},$ that imposed by the robot, and the required minimum time $f.$ The desired position vector, $\mathbf{p}_d(t) = [x_d(t), y_d(t)]^T,$ the velocity vector, $\mathbf{v}_d,$ and the acceleration vector, $\mathbf{a}_d,$ of any point along this segment can be determined in a vector form using (3.1) and (3.2) as follows:

$$\mathbf{p}_d(t) = \mathbf{b}t^3 + \mathbf{c}t^2 + \mathbf{d}t + \mathbf{e} \quad (3.3)$$

$$\mathbf{v}_d(t) = 3\mathbf{b}t^2 + 2\mathbf{c}t + \mathbf{d} \quad (3.4)$$

$$\mathbf{a}_d(t) = 6\mathbf{b}t + 2\mathbf{c} \quad (3.5)$$

where $\mathbf{b}, \mathbf{c}, \mathbf{d}$ and \mathbf{e} are coefficient vectors, for example $\mathbf{b} = [b_x, b_y]^T.$

In order to obtain a complex trajectory consisting of multiple segments, the continuity must be guaranteed. Obviously the continuity conditions between two segments can be ensured by equating the initial conditions of the new segment to the final conditions of the previous segment.

3.3 Trajectory Tracking Controllers

The task of the trajectory tracking controller is to compute the appropriate robot commands so that the robot can track the reference trajectory points. Those reference points are defined as the desired robot states at each time step given by $\mathbf{q}_d(t) = [\mathbf{p}_d(t)^T, \theta_d(t)]^T$, in which $\mathbf{p}_d(t)$ and $\theta_d(t)$ are the desired position vector and the desired robot orientation, respectively. From those, the desired robot orientation $\theta_d(t)$ and the linear and angular velocities, $v_d(t)$ and $\omega_d(t)$ respectively can be obtained as follows. Only for the ideal case, the robot commanded by the desired velocities can track a trajectory with no error. So, mainly the trajectory tracking controller consists of feedforward part and an error feedback part. The more details about control systems can be found in [64].

$$\theta_d(t) = \text{atan2}(\dot{y}_d(t), \dot{x}_d(t)) + k\pi, \quad (3.6)$$

$$v_d(t) = \pm \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (3.7)$$

$$\omega_d(t) = \frac{\dot{x}_d(t)\ddot{y}_d(t) - \dot{y}_d(t)\ddot{x}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \quad (3.8)$$

where $k = 0, 1$, in which $k = 0$ is considered for the forward motion while $k = 1$ is considered for the backward motion. The positive and the negative signs of the desired linear velocity determine the forward and the backward motion of the robot, respectively.

3.3.1 Feedforward Controller

Feedforward controller is an open loop control system that uses directly the desired linear and angular velocities $v_d(t)$ and $\omega_d(t)$ respectively to track the desired planned trajectory. Based on a clockwise turn, and assuming nonholonomic constraints, namely perfect rolling constraints [65], the commanded right and left wheel velocities, $v_r(t)$ and $v_l(t)$, respectively, for this case can be given as follows:

$$v_r(t) = v_d(t) - \frac{W}{2}\omega_d(t) \quad (3.9)$$

$$v_l(t) = v_d(t) + \frac{W}{2}\omega_d(t) \quad (3.10)$$

where W is the wheel base.

3.3.2 Linear State Tracking Controller

Considering a tangent linearization along the reference trajectory, the linear state-tracking controller for the linearized error dynamics can be given as follows. The more details can be found in [52], [56], [57], [66] and [67].

$$v(t) = v_d(t) \cos(\theta_d - \theta_c) + k_1((x_d - x_c) \cos \theta_c + (y_d - y_c) \sin \theta_c) \quad (3.11)$$

$$\omega(t) = \omega_d(t) + \text{sign}(v_d(t))k_2((x_d - x_c) \cos \theta_c - (y_d - y_c) \sin \theta_c) + k_3(\theta_d - \theta_c) \quad (3.12)$$

where $v(t)$ and $\omega(t)$ are the commanded linear and angular velocities for the robot, respectively. x_c , y_c and θ_c are the current X- and Y- positions and the orientation of the robot. k_1 , k_2 and k_3 are the gains of the controller given as follows

$$k_1 = k_3 = 2\xi \sqrt{\omega_d^2(t) + g v_d^2(t)} \quad (3.13)$$

$$k_2 = g |v_d(t)| \quad (3.14)$$

in which $\xi \in (0, 1)$ is the damping coefficient and $g > 0$ is a tuning parameter. The commanded right and left wheel velocities, $v_r(t)$ and $v_l(t)$, respectively, for this case can be obtained by replacing $v_d(t)$ and $\omega_d(t)$ given in (3.9) and (3.10) by $v(t)$ and $\omega(t)$ defined by (3.11) and (3.12), respectively.

3.3.3 Nonlinear State Tracking Controller

The nonlinear design of the state tracking controller is given as follows [52], [56], [57], [66] and [67].

$$v(t) = v_d(t) \cos(\theta_d - \theta_c) + k_1((x_d - x_c) \cos \theta_c + (y_d - y_c) \sin \theta_c) \quad (3.15)$$

$$\omega(t) = \omega_d(t) + \overline{k_2} v_d(t) \frac{\sin(\theta_d - \theta_c)}{\theta_d - \theta_c} ((x_d - x_c) \cos \theta_c - (y_d - y_c) \sin \theta_c) + k_3(\theta_d - \theta_c) \quad (3.16)$$

where $\overline{k_2} = g$. Similarly, the commanded right and left wheel velocities, $v_r(t)$ and $v_l(t)$, respectively, for this case can be obtained by replacing $v_d(t)$ and $\omega_d(t)$ given in (3.9) and (3.10) by $v(t)$ and $\omega(t)$ defined by (3.15) and (3.16), respectively.

Obviously, the feedforward controller suffers from the absence of the feedback action which causes the accumulation of the tracking errors throughout the planned trajectory. Thus inaccurate position is obtained by this method especially for long trajectory. The gains of the feedback action of both the linear and nonlinear state tracking controllers depend on a damping coefficient and a tuning parameter. Those parameters are determined based on a trial and error method. If those parameters are determined for a certain trajectory under certain condition, it may not be suitable for another trajectory with another condition. So, those parameters are required to be determined before the application of the controller. Additionally, those parameters considerably affect the trajectory errors of the methods.

3.4 Pure Pursuit Trajectory Tracking Approach

Pure pursuit method is a simple geometrical based path tracking algorithm. It calculates repeatedly a circular arc that the robot should follow to move from its position to a desired position on the path. The name of this algorithm comes from its behavior, namely it tends to chase a moving point which is a distance ahead.

3.4.1 Review of the Basic Pure Pursuit Path Tracking

The basic pure pursuit path tracking algorithm calculates the curvature of a circular arc that is tangent to the local Y-axis of the robot and connecting the current position of the robot and the goal position (x, y) on the desired path. The goal position is taken as a fixed distance ahead l from the robot position. Using the local coordinates (X, Y) of the robot as shown in Fig. (3.1), the curvature of the circular arc can be calculated and it is then used to calculate the ratio between the commanded wheel speeds. The algorithm is given as follows [68].

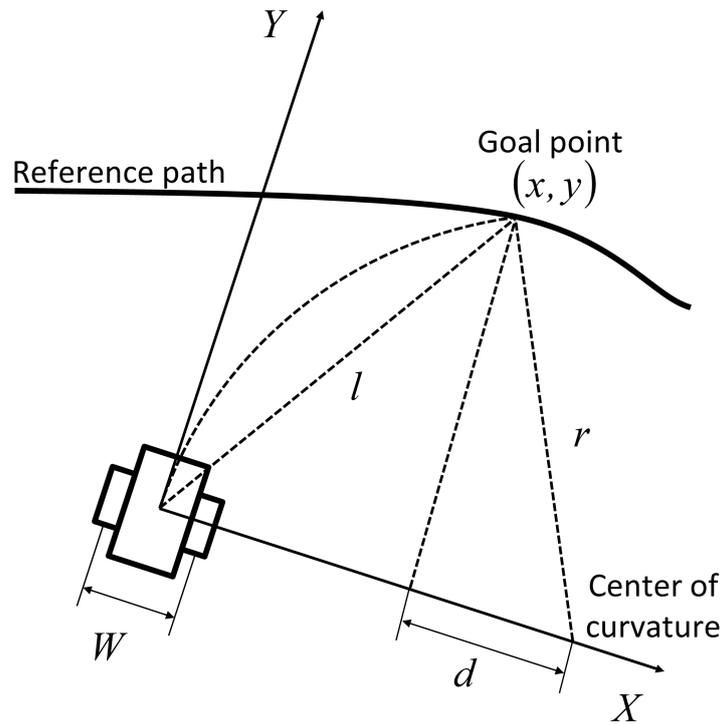


Figure 3.1: The geometry of the algorithm of the original pure pursuit method

$$d = r - x \quad (3.17)$$

where r is the radius of the circular arc connecting the current and the goal positions of the robot.

$$r^2 = d^2 + y^2 \quad (3.18)$$

Substituting (3.18) into (3.17) after squaring both of its sides, and then substituting $x^2 + y^2 = l^2$ into the obtained equation, we get

$$2rx = l^2 \quad (3.19)$$

Thus, the radius of the circular arc r can be obtained as follows:

$$r = \frac{l^2}{2x} \quad (3.20)$$

Then the curvature $\gamma = 1/r$ is given as follows

$$\gamma = \frac{2x}{l^2} \quad (3.21)$$

The term x and $2/l^2$ of the calculated curvature can be interpreted as the error signal and the gain of the control law, respectively.

3.4.2 The Developed Pure Pursuit Trajectory Tracking Approach

Let the robot state at time t be represented as $\mathbf{q}_c(t) = [\mathbf{p}_c(t)^T, \theta_c(t)]^T$ in which the position vector at time t is given by $\mathbf{p}_c(t) = [x_c(t), y_c(t)]^T$ and the orientation of the robot at time t is $\theta_c(t)$. The proposed approach here depends on two input states; namely, the actual state $\mathbf{q}_c(t)$ and the goal state at the n next step of the robot $\mathbf{q}_d(t_n) = [\mathbf{p}_d(t_n)^T, \theta_d(t_n)]^T$ where $t_n = t + n \Delta t$, in which Δt is the sampling time. It is worthy to note that n must be chosen as $n \geq 1$ for keeping the stability of the robot and decreasing the tracking errors drastically. The actual state can be obtained entirely by using the odometry system, while goal state at the n next step can be calculated from the trajectory planner at the given t_n time.

Since the motion of the robot, within a certain sampling time, can be viewed generally as a circular arc with different radius or a straight line as a special case for a circle with infinite radius, then the motion can be realized by the command ‘‘Set speed’’ for the robot. In other words, only within one sampling time, one speed command can be given and accomplished by the robot. Now, consider two position vectors, one is at the planned trajectory, $\mathbf{q}_d(t_n)$, while the other is at the current position vector given by $\mathbf{q}_c(t)$. Then there is only one circular arc that can be visualized to connect those two positions with a tangent at the current one represented by the orientation of the robot at that time, as shown in Fig. (3.2). Thus the arc length, $l_c(t)$, can be given as follows

$$l_c(t) = 2r(t)\theta_b(t) \quad (3.22)$$

where $r(t)$ is the radius of this arc given below and $2\theta_b(t)$ is the arc angle as shown in Fig. (3.2)

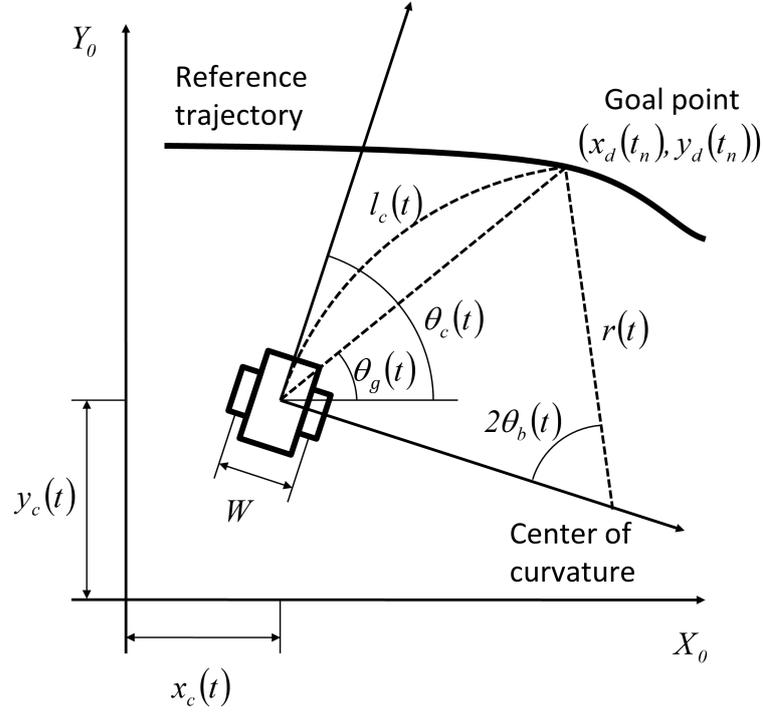


Figure 3.2: The geometry of the algorithm of the developed pure pursuit method

$$r(t) = \frac{\frac{\Delta d(t)}{2}}{\cos(\frac{\pi}{2} - \theta_b(t))} \quad (3.23)$$

$$\theta_b(t) = \theta_c(t) - \theta_g(t_n) \quad (3.24)$$

in which $\Delta d(t) = \|\mathbf{p}_d(t_n) - \mathbf{p}_c(t)\|$ is the Euclidean distance between the desired position and the current one. The angle $\theta_g(t_n)$ is given as follows

$$\theta_g(t_n) = \text{atan2}((y_d(t_n) - y_c(t)), (x_d(t_n) - x_c(t))) \quad (3.25)$$

Using this arc length, the commanded linear speed of the right and left wheels of the robot, $v_r(t)$ and $v_l(t)$ respectively at each sampling time can be calculated as follows:

$$v_r(t) = \frac{(r(t) - \frac{W}{2}) 2 \theta_b(t)}{n \Delta t} \quad (3.26)$$

$$v_l(t) = \frac{(r(t) + \frac{W}{2}) 2 \theta_b(t)}{n \Delta t} \quad (3.27)$$

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

Obviously, if $\theta_b(t)$ is equal to zero or π , then this arc becomes a straight line and $v_r(t)$ and $v_l(t)$ can be simply calculated as follows:

$$v_l(t) = v_r(t) = \begin{cases} \frac{-\Delta d(t)}{n \Delta t} & \text{if } \theta_b(t) = \pi \\ \frac{\Delta d(t)}{n \Delta t} & \text{if } \theta_b(t) = 0 \end{cases} \quad (3.28)$$

Now the scenario of the whole developed pure pursuit trajectory tracking approach for one segment is given as follows:

1. Plan a trajectory using the trajectory planner to get $\mathbf{q}_d(t_n)$ and the minimum time for a trajectory segment (f)
2. While $t \leq f$
 - (a) calculate $\mathbf{q}_d(t_n)$
 - (b) calculate $v_r(t)$ and $v_l(t)$ as described previously
 - (c) command the robot with 'Set Speed' command ($v_r(t), v_l(t)$)
 - (d) update odometry
 - (e) update sampling time
3. end

A block diagram describing the functionality of the developed system is illustrated in Fig. (3.3) where the implementation of the trajectory tracker was changed for comparison in the next section.

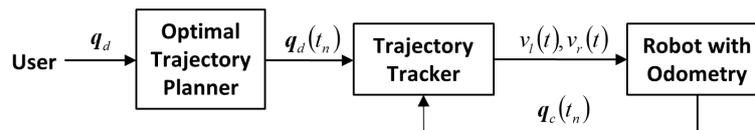


Figure 3.3: The block diagram of the developed approach.

3.5 Experimental Validation and Comparative Study

Experiments have been conducted on the differential-drive mobile robot Koala II [69] manufactured by K-Team which has the following specifications. The maximum velocity and acceleration are $v_{max} = 0.38 \text{ m/s}$ and $a_{max} = 0.075 \text{ m/s}^2$, respectively. The weight of Koala II robot with its battery is 4 kg. This robot is capable of carrying a maximum payload of 3 kg and running continuously for 4 hours.

The setup of the experiments takes place by connecting the Koala II robot with an ASUS Eee PC 1201 PN Nettop with a memory of 2 GB and 1.66 GHz multithreaded CPU, which is held on the board of the robot. The communication is carried out by a USB to Serial adapter with 9600 Baud as shown in Fig. (3.4). The algorithms are MATLAB scripts running on an Ubuntu operating system.



Figure 3.4: The experiment setup of the differential-drive Koala II robot

The reference trajectory is taken as an '∞' shape trajectory starting at point O (0, 0) and passing through points A (0.5, 0.5), B (0.5, -0.5), C (-0.5, 0.5) and D (-0.5, -0.5) in meters and finally ending back at point O (0, 0). This trajectory consists of 5 cubic spline segments. The initial linear velocity of the robot is $v_d(0) = 0 \text{ m/s}$ with orientation $\theta_d(0) = 90^\circ$ and the final one of the first segment at point 'A' is $v_d = 0.1 \text{ m/s}$ with $\theta_d = 0^\circ$. The linear velocity

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

at the end of each segment is $v_d = 0.1 \text{ m/s}$ except at the final segment which is $v_d = 0 \text{ m/s}$. The robot orientation at points 'B' and 'C' is $\theta_d = 180^\circ$, while that at point 'D' is $\theta_d = 0^\circ$. The initial condition of any new segment is taken equal to the final conditions of the previous segment to ensure the continuity conditions between segments. The maximum permissible velocity is taken as $v_{max} = 0.3 \text{ m/s}$ while that of the acceleration is taken as $a_{max} = 0.07 \text{ m/s}^2$. The minimum time required for each segment of the trajectory is calculated online using the procedure described in previous sections. The planned trajectory, the velocity and the acceleration as a function of the time t are also calculated.

Since the basic pure pursuit method is a pure path tracking algorithm which can not consider the execution time and can not satisfy the velocity and the acceleration constraints imposed by the Koala II robot, only the feedforward method, the state tracking methods both linear and nonlinear and the developed pure pursuit trajectory tracking method are used for conducting the experiments. Also, they are only used for the comparison among each others. For the sake of comparison among the feedback controllers, the experiments are conducted five times and the average of the root mean square errors in the X and Y directions and in the orientation are calculated. Moreover, the standard deviation for each is calculated for comparing the reliability of the considered results. The obtained trajectories of each method with the reference trajectory are depicted in one figure. Also, the obtained and the reference linear velocity profiles are shown in another figure.

Before conducting the experiments, the Koala II robot carrying the Nettop on its board is calibrated to eliminate the systematic odometry errors described in [70] yielding the wheel radius to be $R_w = 41.09 \text{ mm}$ and the wheel base to be $W = 306 \text{ mm}$. While conducting the experiments, the actual state of the robot (the current position and orientation) can be obtained entirely by using the odometry system as follows:

$$\Delta\theta_c = 2\pi \frac{R_w}{W} \frac{T_l - T_r}{T_R} \quad (3.29)$$

$$\Delta x_c = R_w \cos(\theta) (T_l + T_r) \frac{\pi}{T_R} \quad (3.30)$$

$$\Delta y_c = R_w \sin(\theta) (T_l + T_r) \frac{\pi}{T_R} \quad (3.31)$$

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

where T_l and T_r are encoder ticks on the left and right wheels, and T_R is the number of encoder ticks in a full rotation of the wheel, that is 5850 in case of the Koala II robot.

The experiments are conducted and the results of a typical execution per each method are presented as follows. The comparison of the trajectories obtained by the different controllers with the reference trajectory is shown in Fig. (3.5). Fig. (3.5A) represents the performance of the four controllers throughout the whole path with the reference path, while Fig. (3.5B) depicts a magnified portion of the whole path to demonstrate the discrepancy of the performance of the four controllers. Obviously, the tracking errors of the feedforward method are accumulating to a high level due to the absence of the feedback action. Due to the feedback mechanism of the linear state tracking controller method, the accumulation of the tracking errors are avoided and then those errors are significantly reduced as shown in Fig. (3.5B). For this method, the tuning parameter and the damping coefficient are chosen by trial and errors as $g = 70$ and $\xi = 0.5$, respectively for the best results. It is remarkably to note that, there is a tradeoff between the stability and the minimum tracking errors regarding the choice of the tuning parameter. The bigger the tuning parameter up to a certain limit, the smaller the errors obtained. Using the same above mentioned tuning parameter and damping coefficient for the nonlinear state tracking controller, the tracking errors are slightly improved. As shown in Fig. (3.5B), the tracking errors of developed pure pursuit controller are significantly improved over the state tracking methods, since its trajectory represented by the solid line is passing through the small circles representing the reference trajectory. There is a tradeoff between the stability and the errors regarding the choice of the n time steps factor developed for this method. Namely, the smaller the n time steps factor the smaller the errors, but the higher the robot oscillations. The time step factor is chosen here by trial and errors to be $n = 4$ time steps ahead for high accuracy tracking errors and more stability with no oscillations. It is remarkably to note that, the n time steps factor has the same effect as that of the lookahead distance developed for the original path tracking method [68].

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

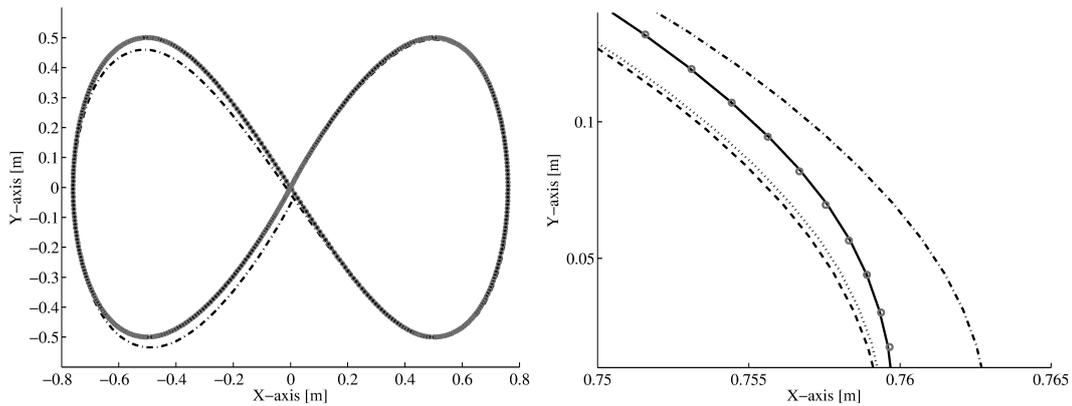


Figure 3.5: The positional tracking errors of the four controllers, (A) the whole path positional tracking errors, (B) a magnified portion of the whole path. The reference (o o o), the feedforward controller (_ . _), the linear state tracking controller (_ _ _), the nonlinear state tracking controller (....), and the pure pursuit trajectory tracking controller (_ _ _) trajectories.

The comparison of the velocity profiles obtained by the different controllers with the reference velocity profile is shown in Fig. (3.6). The discrepancies between the obtained and the reference velocities are reasonable in all of the compared cases. The main purpose of the trajectory tracking controller is to control the state of the robot using the desired state as a function of time. Therefore the velocity profiles contain minor information about the performance of the controller. But it can be noted that, due to the absence of the feedback action, the velocity profile of the feedforward method is typically the reference one while it is slightly deteriorated in the other three controllers and it is shown to be inversely proportional to the tracking errors.

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

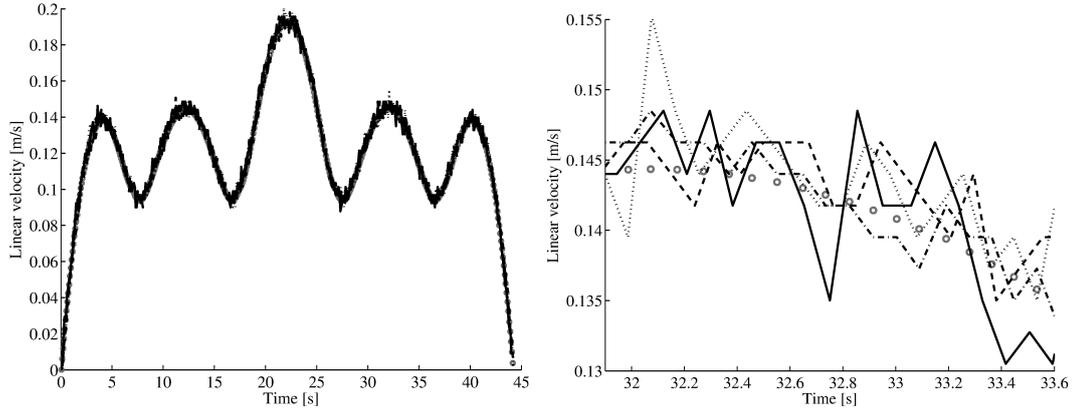


Figure 3.6: The velocity profiles of the four controllers, (A) the whole velocity profiles, (B) a magnified portion of the velocity profiles. The reference (o o o), the feedforward controller ($-\cdot-$), the linear state tracking controller ($---$), the nonlinear state tracking controller (\dots), and the pure pursuit trajectory tracking controller ($---$) profiles.

Since the tracking errors of the feedforward method are dependent of the length of the path for the experiments then only the three feedback tracking controllers are compared, namely the linear and nonlinear state tracking controllers and the developed pure pursuit trajectory tracking method. The root mean square error per each experiment for each of the three methods is calculated as follows:

$$z_{rms} = \sqrt{\frac{\sum (z_d(t) - z_c(t))^2}{m}} \quad (3.32)$$

where $z_d(t)$ and $z_c(t)$ represent each of the desired and the current position and orientation, namely X, Y and θ values, while m is the total number of sampling time. Then the average and the standard deviation of the five experiments per each method are calculated as follows for the comparison of those methods.

$$\bar{z} = \frac{1}{5} \sum z_{rms} \quad (3.33)$$

$$\sigma = \sqrt{\frac{1}{5} \sum (z_{rms} - \bar{z})^2} \quad (3.34)$$

The comparison results are shown in Fig. (3.7), in which the average and standard deviation of the X and Y tracking errors are shown in Fig. (3.7A), while that of the orientation

error is shown in Fig. (3.7B). The nonlinear state tracking controller is slightly better than the linear one in terms of average error and the reliability represented by the standard deviation. The tracking errors of the pure pursuit tracking controller significantly improved over the other two methods, namely, more than six folds improvement in the average tracking errors in the X and Y directions, while almost five folds improvement of the orientation errors take place. It is remarkably to note that, the decrease of the standard deviation of the developed pure pursuit method shows increased reliability.

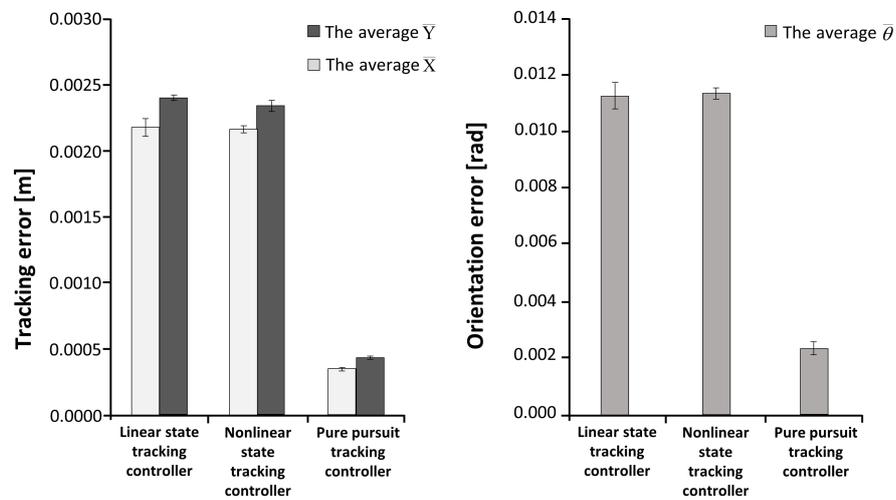


Figure 3.7: The average and the standard deviation of the linear and the orientation errors

3.6 Benefits of Pure Pursuit Trajectory Tracking

The trajectory planning and tracking for the differential-drive mobile robots have been presented in this Chapter and the experiments have been conducted on the Koala II robot. The trajectory planning was achieved using the cubic spline that interpolates waypoints and satisfies the velocity and the acceleration constraints imposed by the robot. A pure pursuit trajectory tracking approach has been developed in this work, in which the original fixed lookahead distance of the method is radically changed to be an updating time dependant lookahead distance which is related to the sampling time. The developed approach enables considering the execution time and the velocity and the acceleration constraints. A com-

CHAPTER 3. PURE PURSUIT TRAJECTORY TRACKING

parison with the commonly used linear and nonlinear state tracking controllers has been studied to bear out the efficacy of the developed method. Significant tracking error reductions and a high reliability have been obtained by the developed method which outperforms the other tracking controllers. The developed method also avoids the selection of the tuning and damping parameters used by the commonly used state tracking controllers. Many feedback systems other than the odometry system used here can be applied with the developed trajectory tracking approach, such as the GPS with the compass enabling wide range of applications.

Chapter 4

Optimized Fuzzy Controller Properties and Applications

4.1 Introduction to Fuzzy Control

Fuzzy control is a control process based on fuzzy logic. It operates with analog values throughout the whole computational cycle using fuzzy membership functions instead of binary values. The two main type of fuzzy controllers are the Mamdani and the Takagi Sugeno types. The main difference between them is the output, where in the Mamdani case fuzzy membership functions are defined, while in the Sugeno case a polynomial function. From now, only the Mamdani type will be discussed.

To describe how a fuzzy controller operates, two different types of questions must be addressed. The first question is 'What to specify before the operation?'. This specification contains everything that is essential to the structure of the controller. The other question involves everything related ot 'How will a fuzzy controller work?'. This is basically the operation protocol of the controller.

4.1.1 Structure of a Fuzzy Controller

Similarly to other controller types, an essential part is the connection with the outside world through inputs (X) and outputs (Y). The communication is carried out by real numbers, thus simplifying the controller integration into a bigger system.

A fuzzy linguistic variable or membership function translates an exact input (x) into the world of fuzzy. A general membership function is the following: $\mu : X \rightarrow [0, 1]$, which represents the truth of a given $x \in X$ for a given μ membership function. If an input x is well represented by μ , then μ will map x around 1 and also if x is not a typical member of μ , then the result will be near 0. On the output of a Mamdani type fuzzy controller, similar membership functions are declared to define the final output membership function of the controller, which represents the overall output.

In the heart of the fuzzy controller, there are rules connecting the inputs with the outputs. A fuzzy rule base is a set of 'if-then' rules.

Currently, fuzzy control approach is mainly concerned with model-based methods. The main idea of reasoning is the modus ponens for forward reasoning and modus tollens for backward reasoning, which are referred as inference in the literature. The kernel of a linguistic fuzzy model [71, 72] is the rule base consisting of r rules with the following form:

$$IF \ x_1 \ IS \ P_s^1, x_2 \ IS \ P_s^2, \dots, x_m \ IS \ P_s^m \ THEN \ y \ IS \ Q_s,$$

where P_s^l (resp. Q_s) are linguistic values of variable X_l (resp. Y), and $x = [x_1, x_2, \dots, x_m]$ are the input values ($s = 1, \dots, r$ and $l = 1, \dots, m$).

4.1.2 Operation of a Fuzzy Controller

To operate a fuzzy controller, the defined rule base has to be evaluated, with the previously introduced 'IF ... THEN ...' form of the rules. The 'IF ...' part is also known as the antecedent, while the 'THEN ...' part is called consequent. During the operation, first all r rules antecedents must be evaluated. An antecedent consists of one or more membership functions combined with fuzzy operations, usually 'AND', 'OR', or 'NOT'. Since the classical imple-

mentation of these operations are min , max , and $(1 - x)$, this task is usually computationally efficient. As a result, the fitness or the truth value of each rule is calculated for any given x input.

This truth value is propagated to the consequents during the implication operation. The classical implementation of it can be carried out by the min function. The result of this calculation is the rule base r consequents with the assigned fitness values of the corresponding antecedents. In theory, the consequents can also be defined as a complex combination of membership functions but in practice, they usually consists of only one such function.

The next step is to generate the output by a complex membership function ($A(y)$) using the previously gathered implied consequents. This process is classically called as aggregation, and usually implemented by the max mathematical function.

The final step is to calculate the most representing element (y^*) of the output membership function. This process gives a real value for each Y output, hence the name of the operation is called defuzzification. One of the best defuzzification technique, therefore the most widely used one is the center of gravity (COG), which can be obtained by calculating the weighted integration. In case of general membership functions, this can only be approximated numerically which requires that both the implication and the aggregation must be carried out numerically as well. As a result, these steps are computationally intensive and only approximations.

4.1.3 Introduction to Defuzzification

As most modeling and control applications require crisp outputs, the fuzzy inference system output $A(y)$ has to be also converted into a crisp output of y^* . This operation is called defuzzification. The most popular defuzzification methods are the COG and the mean of maxima (MOM) methods.

More general frameworks have been proposed in which the COG and MOM defuzzification methods have their place, such as the parametric basic defuzzification distribution (BADD) and semi-linear defuzzification (SLIDE) methods of Yager and Filev [73, 74]. They

are essentially based on the transformation of a possibility distribution into a probability distribution based on Klir's principle of uncertainty invariance. The main emphasis is the learning of the parameters involved, which is treated as an optimization problem [75–77].

Note that in the literature the terms for describing the different defuzzification methods vary from source to source. The terms 'center of gravity' defuzzification, 'center of area' defuzzification and 'center of sum' defuzzification, for instance, refer to different methods in some sources and are used as synonyms in others. Therefore, we should pay attention to the formal definitions of the defuzzification methods rather than to their names. In [78, 79], comprehensive overviews are given for defuzzification methods. Here, the same terminology is used as in [79].

In the article [80], the authors present two computational methods. The slope-based and the modified transformation function methods are introduced for the center of gravity defuzzification for trapezoidal membership functions forming a fuzzy partition.

This work focuses on the computational aspects of an arithmetic based defuzzification method. When applying the new method to a properly prepared fuzzy control system, the crisp output y^* will change continuously as the input values vary continuously, which is a desirable property in modeling and control applications. However, the most popular COG defuzzification method has a high computational burden [81, 82], which is a significant disadvantage in control and model identification, and in tuning applications. This high cost is often circumvented by introducing new defuzzification methods that seek to approximate the center of gravity [82, 83].

4.1.4 Classical Defuzzification Methods

The classical defuzzification techniques are usually among the following methods.

The COG defuzzification is the most widely implemented approach mainly because it produces a very smooth output. To calculate this, the following integration should be determined in a numerical way which is computationally intensive. The output of the COG can be expressed with the following formula:

$$z_0 = \frac{\int \mu_i(x)xdx}{\int \mu_i(x)dx}$$

where z_0 is the defuzzified output, μ_i is a membership function and x is output variable.

The bisector (BIS) defuzzification simply divides the aggregated membership function area into two equal areas at point z_0 :

$$\int_{\alpha}^{z_0} \mu_A(x)dx = \int_{z_0}^{\beta} \mu_A(x)dx.$$

The MOM defuzzification method calculates the average value of the locations of the maximum value of the aggregated membership function:

$$z_0 = \frac{\int_{x'} xdx}{\int_{x'} dx'}$$

where $x' = (x, \mu_A(x) = \max(\mu_A))$.

The smallest of maxima (SOM) defuzzification method can be calculated as the following:

$$z_0 = \min(x')$$

where $x' = (x, \mu_A(x) = \max(\mu_A))$.

The largest of maxima (LOM) defuzzification method can be calculated as the following:

$$z_0 = \max(x')$$

where $x' = (x, \mu_A(x) = \max(\mu_A))$.

4.2 Theoretical Background

The theoretical background of the proposed fuzzy arithmetic and defuzzifications are based on the work of József Dombi.

A simplified version of a fuzzy rule base is similar to an implication system such as the following:

$$\begin{aligned}
 &\text{if } x_1 \text{ then } y_1 \\
 &\text{if } x_2 \text{ then } y_2 \\
 &\dots\dots\dots \\
 &\text{if } x_n \text{ then } y_n.
 \end{aligned}
 \tag{4.1}$$

Our first observation is that (4.1) is an implication system describing when x_i appears and then y_i also appears i.e. the two events are coupled to each other. We are convinced that a similar case is also true in the fuzzy control system. Our second observation is that a continuous-valued logical expression is in the fuzzy control after 'if' which expresses a numerical value instead of an uncertain condition.

Let $\mathcal{L}_i(\mathbf{x})$ be the i -th logical expression then a rule system can be formalized in the following way:

$$\begin{aligned}
 &\text{if } \mathcal{L}_1(\mathbf{x}) \text{ then } y_1 \\
 &\text{if } \mathcal{L}_2(\mathbf{x}) \text{ then } y_2 \\
 &\dots\dots\dots \\
 &\text{if } \mathcal{L}_n(\mathbf{x}) \text{ then } y_n.
 \end{aligned}
 \tag{4.2}$$

The application of this rule system if \mathbf{x}^* is the input and $\mathcal{L}_i(\mathbf{x}^*) = \alpha_i$ gives:

$$\begin{aligned}
 &\alpha_1, y_1 \\
 &\alpha_2, y_2 \\
 &\dots\dots\dots \\
 &\alpha_n, y_n.
 \end{aligned}
 \tag{4.3}$$

But what is α_i ? The meaning of α_i is the 'probability' and 'applicability' of the i -th rule.

We can normalize α_i in the following way:

$$w_i = \frac{\alpha_i}{\sum_{i=1}^n \alpha_i}, \text{ if } \sum \alpha_i \neq 0.
 \tag{4.4}$$

In case of $\sum_{i=1}^n \alpha_i = 0$, then we can not infer anything.

The output in this case is

$$y^* = \sum_{i=1}^n w_i y_i. \quad (4.5)$$

1. The value of y^* is the generalized arithmetic mean of y_i -s and the weights are the truth values of the antecedents.
2. We can interpret y^* as an expected value, where y_i -s are the values and w_i -s are the probabilities which are determined by the rules and express its probability or applicability.
3. In fuzzy control, y^* can be interpreted as the COG according to the terminology used in Mamdani defuzzification.

We can see in the above-mentioned method that there is no room for implication. We only use arithmetic operations! Now let us turn to the Mamdani model. The only change that we can make is, that the consequence part is not a value as in (4.1) and (4.2) but fuzzy sets i.e. y_i are membership functions. To generalize the procedure, the weighted arithmetic mean of the membership functions has to be calculated. In case of choosing either triangular or trapezoid functions, this calculation becomes very simple.

In the presented ABC method, the linear combination of the consequent of the rules will be used where the weights are the normalized values of the antecedents.

To achieve this, first an effective algorithm needs to be developed to calculate the linear combination of the membership functions.

4.3 Applied Fuzzy Arithmetics

The suggestion that fuzzy quantities could be arithmetically combined according to the laws of fuzzy set theory is due to Zadeh [84]. Soon after, several researchers worked independently along these lines, like Jain [85], Mizumoto and Tanaka [86, 87], Nahmias [88], Nguyen [89], Dubois and Prade [90]. It was only later realized that the mathematics of fuzzy

quantities are an application of possibility theory, an extension of interval analysis as well as of the algebra of many-values quantities (Young [91]). Fuzzy interval theory extends and updates the overview of Dubois and Prade [92]. Theoretical details and applications can be found e.g. in monographs by Kaufmann and Gupta [93, 94], and Mares [95]. In 1987, there was a special issue of Fuzzy Sets and Systems. Dubois and Prade [96] focused on the fuzzy intervals domain, and more recently another one has appeared (Fullér and Mesiar [97]). Fuzzy arithmetic-based α -cuts are where the result of an α -cut is represented by an interval. The arithmetics can be understood as an interval arithmetics of the α -cuts. Instead of dealing with intervals we deal with left and right-hand sided soft inequalities that define the interval. A new calculation procedure of arithmetic is presented, where these soft inequalities have certain properties (i.e. the inequality is represented by strict monotonously increasing function). Using this technique, the result of linear combinations will also be linear (i.e. it is closed under linear combination). The soft inequalities define an interval by using proper conjunctive and disjunctive operator. In real world applications, we often need to deal with imprecise quantities. They may be the results of measurements or vague statements, e.g. 'I have about 40 dollars in my pocket', 'she is approximately 170 cm tall'. In fuzzy arithmetics we can use the $a < x$ and $x < b$ inequalities to characterize such quantities. E.g. if I have about 40 dollars I probably have more than 35 dollars but less than 45 dollars. Fuzzy numbers can also be used to represent imprecise quantities. Fuzzy numbers are created by *softening* the $a < x$ and $x < b$ inequalities, i.e. replacing the crisp characteristic function with two fuzzy membership functions and applying a fuzzy conjunction operator to combine the two functions. We refer to the softened inequalities as *fuzzy inequalities*. We call the distending function corresponding to the $x < a$ interval the left-hand side of the fuzzy number and denote it by δ_l . Similarly, we refer to the distending function corresponding to the $x < b$ interval as the right-hand side of the fuzzy number and denote it by δ_r . Here, the following terminology will be used. A function representing the soft inequalities are called the distending function. Naturally one would like to execute arithmetic operations over fuzzy numbers. Fuzzy arithmetic operations are generally carried out using the α -cut method. Fuzzy arith-

metic operations are based on the extension principle of arithmetic. In arithmetic theory we can find the result of an arithmetic operation by measuring the distance of the operands from the zero point, then apply the operation on these distances. We can extend this calculation to intervals.

In fuzzy arithmetic theory we deal with fuzzy numbers which are mappings from real numbers to the $[0,1]$ real interval. Operations are executed by creating an α -cut for all $\alpha \in [0,1]$ and using the interval arithmetic principle to get the resulting value for each α value. Fig. (4.1) shows fuzzy addition with fuzzy numbers represented as a line. The rightmost triangle number is the sum of the two other triangles.

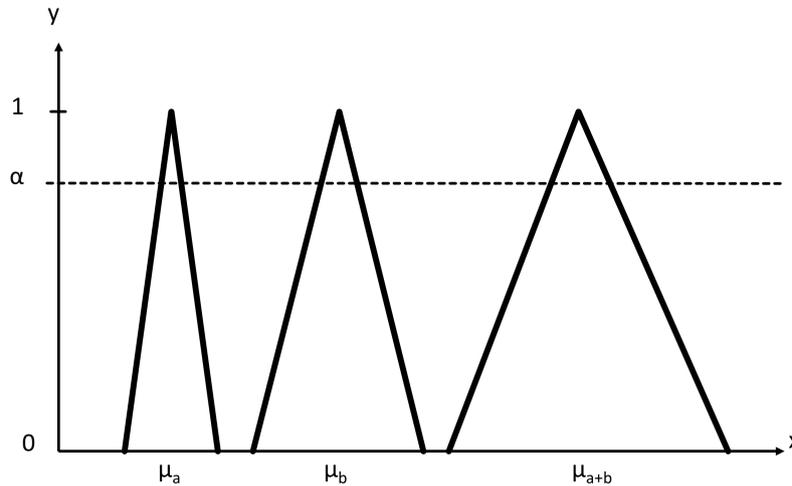


Figure 4.1: Fuzzy addition with α -cut.

This way, all the well-known unary $(-x, x^2)$ and binary operations $(x + y, xy)$ can be obtained as fuzzy operations. However, the calculation of fuzzy operations with the α -cut is tedious and often impractical. Here, a new efficient method is presented which is equivalent to the α -cut. Fuzzy numbers are often composed of two strictly monotone functions, i.e. the left-hand side denoted by μ_l , and the right-hand side denoted by μ_r of the fuzzy number. Fuzzy operations can be carried out by first applying them to the left hand sides then to the right hand sides of the operands. This separation allows us to treat fuzzy numbers as strictly monotone functions when dealing with fuzzy arithmetic operations. In the following, we omit the subscript from μ_l and μ_r and simply write μ with the inherent assumption that

we shall only do arithmetic operations with functions representing the same side of fuzzy numbers.

4.3.1 Arithmetics based on Inverse Functions

Distending function is a measurement that express the truth of an inequality, i.e.

$$\delta_a(x) = \text{truth}(a < x).$$

In this case, the following method is equivalent to the α -cut.

Let $\mu_1, \mu_2, \dots, \mu_n$ ($n \geq 1$) be strictly monotone functions representing fuzzy inequalities and let F be an n -ary fuzzy operation over them. If

$$\mu = F(\mu_1, \mu_2, \dots, \mu_n),$$

then

$$\begin{aligned} \mu(z) &= (F(\mu_1^{-1}, \mu_2^{-1}, \dots, \mu_n^{-1}))^{-1}(z) \\ \mu(z) &= \sup_{F(x_1, x_2, \dots, x_n)=z} \min\{\mu_1(x_1), \mu_2(x_2), \dots, \mu_n(x_n)\}. \end{aligned} \tag{4.6}$$

Fig. (4.2) shows equivalence for the addition of lines. The left hand-side shows the result of lines added together using the α -cut presented as dashed line. On the right hand-side, the inverse functions of the two operands were simply added together. The result is also represented by a dashed line. It can be seen from the figures that the result of the α -cut is indeed the inverse of the result on the right hand-side.

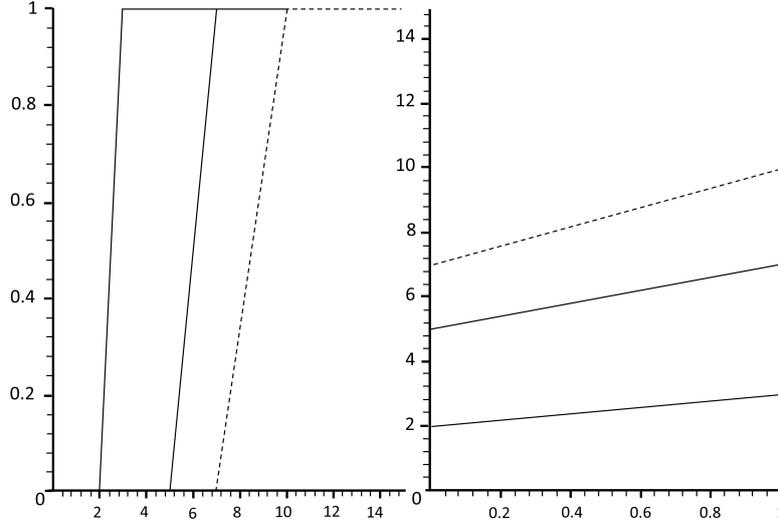


Figure 4.2: Left: α -cut addition, right: inverse of addition.

We can state the following for the properties of fuzzy operations by using equation (4.2). Let $\mu_1, \mu_2, \dots, \mu_n$ ($n \geq 1$) be strictly monotone functions representing fuzzy inequalities and let F be an n -ary fuzzy operation over them. If

$$F(\mu_1^{-1}, \mu_2^{-1}, \dots, \mu_n^{-1})$$

is strictly monotone, then F has all the properties as its non-fuzzy interpretation.

Calculation of the Linear Combination

We suppose that the consequent is a triangle or a trapezoid membership function.

The equations of the left and right hand-sides have the following forms:

$$\begin{aligned} y &= m_l(x - a_l) + \frac{1}{2} \\ y &= m_r(x - a_r) + \frac{1}{2} \end{aligned} \tag{4.7}$$

where $m_e > 0$, $m_r < 0$, $-\infty < a_e < a_r < \infty$. See Fig. (4.3)

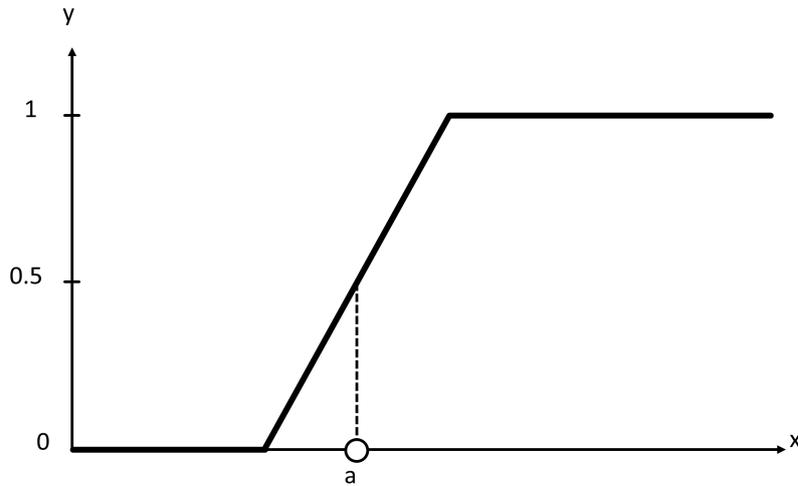


Figure 4.3: Line given by its mean value a and tangent $m = \tan \alpha$.

If the trapezoid is rectangle then $m = \infty$ and the equation is

$$y(x) = \begin{cases} \frac{1}{2}, & \text{if } x = a \\ 0, & \text{if } x < a \\ 1, & \text{if } x > a \end{cases} \quad (4.8)$$

See Fig. (4.4)



Figure 4.4: One-sided open interval

The interpretation of (4.7) and (4.8) are $truth(a < x)$ i.e. the membership function is a soft inequality.

Triangle fuzzy numbers are commonly used to represent approximate values. A triangle fuzzy number has one line on each side. We can add triangle fuzzy numbers by first adding their left hand lines and then add their right hand lines together. This method can be generalized too.

A line $l_a^{(m)}(x)$ is given by its mean value if:

$$l_a^{(m)}(x) = \max \left(\min \left(m(x-a) + \frac{1}{2}, 0 \right), 1 \right) = [a <_m x],$$

as shown in Fig. (4.3).

Here we introduce a new notation:

$$[x] = \max \left(0, \min(x, 1) \right) = \begin{cases} 1, & \text{if } x > 1 \\ x, & \text{if } 0 < x < 1 \\ 0, & \text{if } x < 0 \end{cases}$$

The following properties can also be seen in Fig. (4.3).

if $a < x$ then $[a <_m x] > \frac{1}{2}$,

if $a = x$ then $[a <_m x] = \frac{1}{2}$,

if $a > x$ then $[a <_m x] < \frac{1}{2}$.

Interpretation of $l_a^{(m)}(x)$ is

$$l_a^{(m)}(x) = truth(a <_m x) = [a <_m x].$$

The inverse of $l_a^{(m)}(x)$ denoted by $l^{-1}(y)$ can be calculated using the formula:

$$l^{-1}(y) = \frac{y - \frac{1}{2}}{m} + a.$$

When we apply an arithmetic operation to fuzzy inequalities we need to ensure that the operation is meaningful, i.e. the inequalities must represent the same sides of the fuzzy numbers. The following criteria encapsulates this requirement.

$$\text{sgn}(m_1) = \text{sgn}(m_2) = \dots = \text{sgn}(m_n).$$

Addition Based on Inverse Function

Let $l_i(x) = m_i(x - a_i) + \frac{1}{2}$ ($i \in \{1, \dots, n\}$) be lines given by their mean values. The fuzzy sum of l_i lines denoted by l is also a line and can be defined as

$$l(x) = l_1(x) \oplus \dots \oplus l_n(x) = m(x - a) + \frac{1}{2},$$

where

$$\frac{1}{m} = \sum_{i=1}^n \frac{1}{m_i} \quad \text{and} \quad a = \sum_{i=1}^n a_i.$$

The following is also true:

$$\begin{aligned} l^{-1}(y) &= (l_1^{-1}(y) + \dots + l_n^{-1}(y)) \\ &= \sum_{i=1}^n \left(\frac{y - \frac{1}{2}}{m_i} + a_i \right) = \sum_{i=1}^n \left(\frac{y - \frac{1}{2}}{m_i} \right) + \sum_{i=1}^n a_i = \\ &= \left(y - \frac{1}{2} \right) \sum_{i=1}^n \frac{1}{m_i} + \sum_{i=1}^n a_i \end{aligned}$$

From this, we find that:

$$l(x) = \frac{1}{\sum_{i=1}^n \frac{1}{m_i}} \left(x - \sum_{i=1}^n a_i \right) + \frac{1}{2}.$$

Substituting $\frac{1}{m}$ and a into the equation we get the desired result. Namely,

$$l(x) = m(x - a) + \frac{1}{2}.$$

Subtraction Based on Inverse Function

Calculations for a subtraction yields the following

$$l(x) = l_1 \ominus l_2 = \frac{1}{\frac{1}{m_1} - \frac{1}{m_2}} (x - (a_1 - a_2)) + \frac{1}{2}.$$

Note that l does not exist when $m_1 = m_2$.

The result of the operation is also a line, i.e. the operation is closed for lines.

Multiplication by a Constant Based on Inverse Function

The following statement is also valid:

Let $l(x) = m(x - a) + \frac{1}{2}$. The multiplication of this line by w we get $l(x) = m^*(x - a^*) + \frac{1}{2}$, where $m^* = \frac{m}{w}$ and $a^* = aw$.

Linear Combination Based on Inverse Function

The linear combination of the left (right) hand sides is linear. The result is:

$$y = m(x - a) + \frac{1}{2}, \quad (4.9)$$

where

$$a = \sum w_i a_i, \quad (4.10)$$

and

$$\frac{1}{m} = \sum_{i=1}^n \frac{w_i}{m_i}. \quad (4.11)$$

4.4 Arithmetic Based Defuzzification Method

The linear combination of a triangle or a trapezoid function is also a triangle or a trapezoid function.

We have seen that the linear combinations of the left and right-hand sides of the membership functions are also linear functions. Let the intersection of this function be at x_1, y_1 . The followings must be proved: $y_1 > 0$.

Let $\mu_A(x_{A_1}) = \alpha$, $\mu_A(x_{A_2}) = \alpha$, and $x_{A_1} < x_{A_2}$, and let $\mu_B(x_{B_1}) = \alpha$, $\mu_B(x_{B_2}) = \alpha$ and $y_{B_1} < y_{B_2}$.

$\mu(x_1^*) = \alpha$, $\mu(x_2^*) = \alpha$, where

$$x_1^* = w_1 x_{A_1} + w_2 y_{B_2} < w_1 x_{B_1} + w_2 x_{B_2} = x_2^*$$

After calculating the linear combination of the left and right-hand sides of the resulting (aggregated) trapezoid, we get the following cases:

- The intersection is above 1, see Fig. (4.5),
- The intersection is below 1, see Fig. (4.6),
- The intersection is equal 1, see Fig. (4.7),

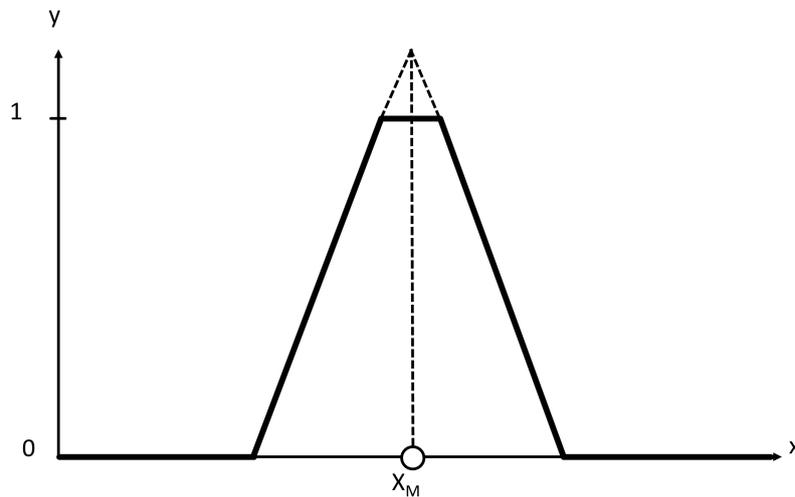


Figure 4.5: Trapezoid membership

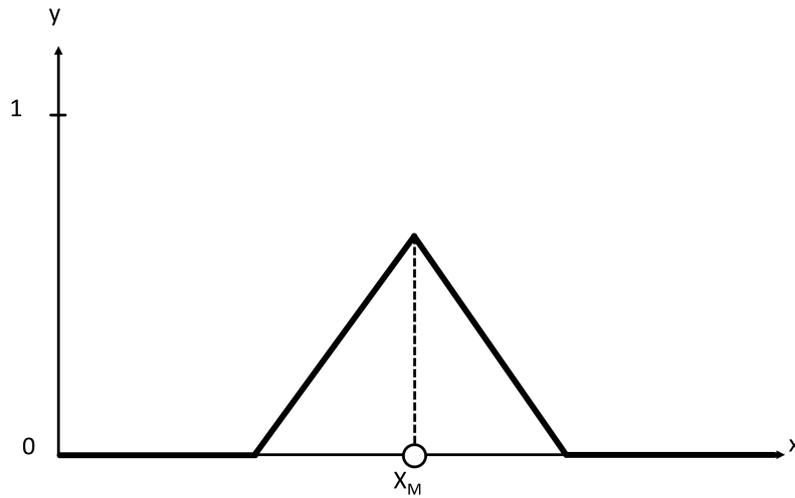


Figure 4.6: Triangle function (not normalized) $\max \mu(x) < 1$

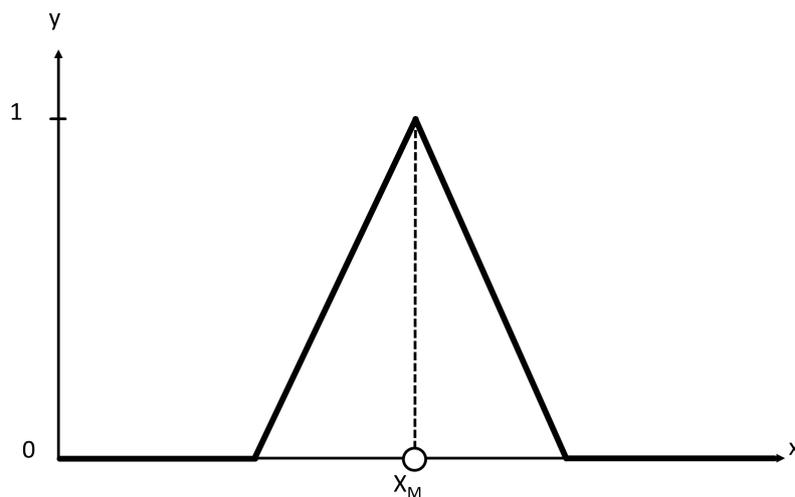


Figure 4.7: Triangle (normalized) i.e.: $\max \mu(x) = 1$

It is also clear, that the intersection cannot be below zero.

The main advantage of using the arithmetic concept is that the result generalizes a trapezoid like function. In our model, we use a new type of defuzzification method. Two defuzzification methods are proposed.

1. Let x_M the intersection of the aggregated left and right-hand sides. Intuitively, it

is the highest probability value of the trapezoid.

$$x_M = \frac{m_1 a_1 - m_2 a_2}{m_1 - m_2}. \quad (4.12)$$

See Fig. (4.5, 4.6, 4.7).

2. Let x_S the center point of the aggregated trapezoid. Intuitively, it is the most stable point of the trapezoid.

$$x_S = \frac{1}{6}(c_1 - c_2) \frac{c_1 + 3b + c_2}{c_1 + 2b + c_2}. \quad (4.13)$$

The meaning of a , b , c_1 , c_2 , x_S can be seen in Fig. (4.8).

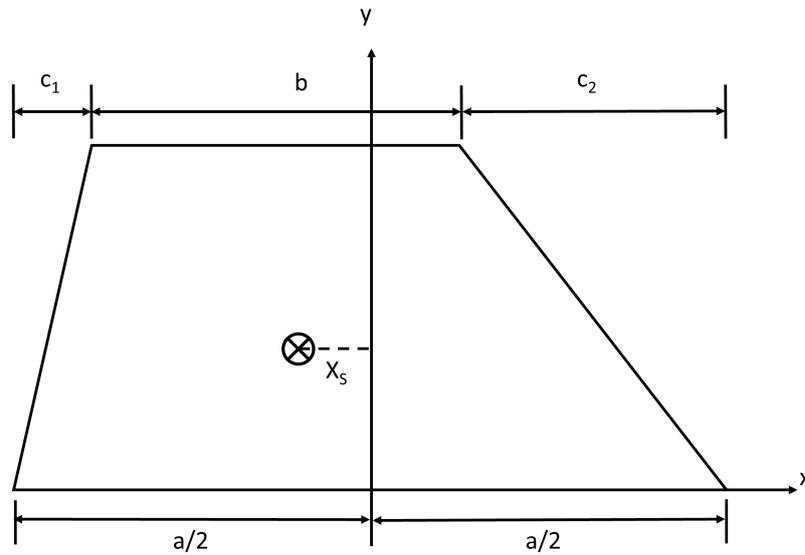


Figure 4.8: Stable center of the trapezoid

The Pliant Model

A Pliant inequality is given as a sigmoid function, that can also express the truth of an inequality.

$$\{a <_{\lambda} x\} = \frac{1}{1 + e^{-\lambda(x-a)}} = \sigma_a^{(\lambda)}(x)$$

where a is the mean value, i.e. $\sigma_a^{(\lambda)}(a) = \frac{1}{2}$ as seen in Fig. (4.9).

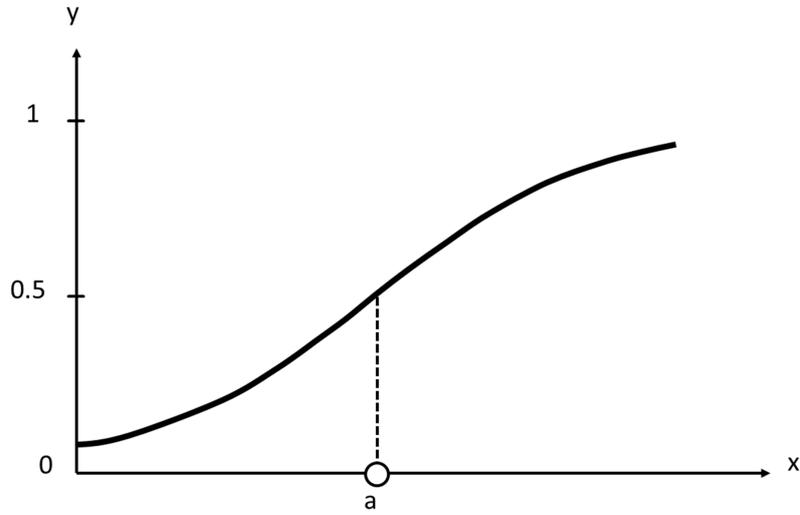


Figure 4.9: Pliant inequality

The following properties can also be seen in Fig. (4.9)

$$\text{If } a < x \text{ then } \{a <_{\lambda} x\} > \frac{1}{2}$$

$$\text{If } a = x \text{ then } \{a <_{\lambda} x\} = \frac{1}{2}$$

$$\text{If } a > x \text{ then } \{a <_{\lambda} x\} < \frac{1}{2}$$

The inverse function of $\sigma_a^{(\lambda)}(x)$ is denoted as $(\sigma_a^{(\lambda)})^{-1}(x)$, and can be calculated as follows.

Let

$$\sigma_a^{(\lambda)}(x) = \frac{1}{1 + e^{-\lambda(x-a)}} = \omega$$

then

$$1 = \omega(1 + e^{-\lambda(x-a)}) = \omega + \omega e^{-\lambda(x-a)}$$

$$\frac{1 - \omega}{\omega} = e^{-\lambda(x-a)}$$

$$\ln\left(\frac{1 - \omega}{\omega}\right) = -\lambda(x - a)$$

$$x = (\sigma_a^{(\lambda)})^{-1}(\omega) = \frac{1}{-\lambda} \ln\left(\frac{1 - \omega}{\omega}\right) + a$$

Other important properties and operations for the Pliant inequalities are detailed in [98]. From now, we assume that a Pliant function can be easily used in a fuzzy controller as a membership function with efficient calculation techniques.

4.5 Advantages of the Arithmetic Based Method

In the following sections, three fuzzy controllers were implemented and compared. All versions rule fitting technique remained the same as originally defined by Zadeh, while the implication, aggregation, and defuzzification methods were modified for the Pliant and ABC cases.

1. Classical fuzzy controller

This method utilizes the well known implication and aggregation methods based on the numerical representation of the output membership functions. The classical defuzzification was the COG method.

2. ABC fuzzy controller

The previously introduced arithmetic based concept was implemented to handle the consequent part of each rule, while the output membership functions were limited to be trigonometric or trapezoid ones.

3. Pliant fuzzy controller

The previously introduced arithmetic based concept was implemented to handle the consequent part of each rule, while the output membership functions were limited to be Pliant functions.

For the experimental comparison, a classical control demo was used to highlight the improvement of the ABC method. The task to be solved consisted of a water tank with one pipe at the bottom and another at the top. The upper pipe is also equipped with a controllable valve, which is capable to increase the water level of the tank, while the pipe at the bottom

only serves as an exhaust. The purpose of the control is to maintain the desired water level in the tank by opening and closing the valve. This demo can be found in the Matlab Fuzzy Logic Toolbox. This controller is a classical Mamdani type. The output was examined with different numerical resolution. We observed significant degradation when the resolution decreased on the other hand the speed is slightly increased. The classical fuzzy controller was compared with the new ABC method as it is shown in Fig. (4.10).

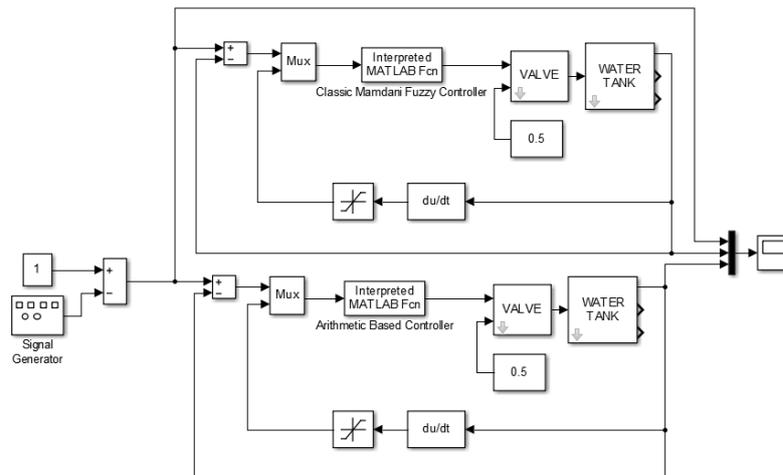


Figure 4.10: Matlab simulink model for Mamdani and ABC method

The detailed analysis of the results can be found in the next sections.

4.5.1 Range independence

The range dependency is a general problem in case of classical Mamdani method with COG defuzzification and often causes designing error. For example, this error occurs when the center of the triangular membership function is defined at the edge of the range. In case of the ABC method, the output is not affected by the range, therefore the triangle will be precisely defuzzified. In Fig. (4.11), the result of ABC method (●) and Mamdani method (○) is depicted.

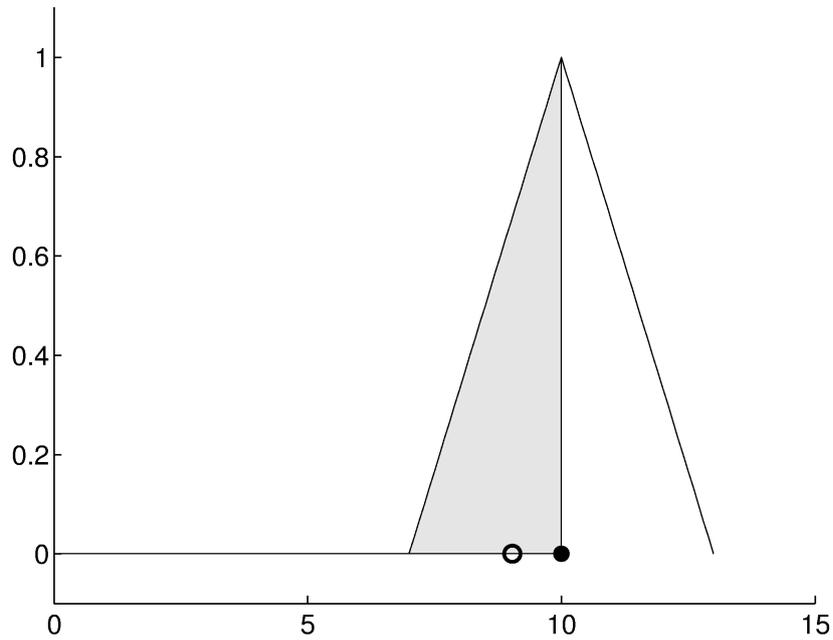


Figure 4.11: Range independence

In the classical Mamdani method, a controversial case can also occur when two membership functions are separated on the output. The classical defuzzification method depends on the range of the control variable as seen in Fig. (4.12) where one of the membership functions is partially outside of the output range. The result of the ABC method is denoted by \bullet and the Mamdani COG method by \circ , and the aggregated output function by the gray area. The central trapezoid (dashed line) is the linear combination of the two output membership functions used by the ABC method.

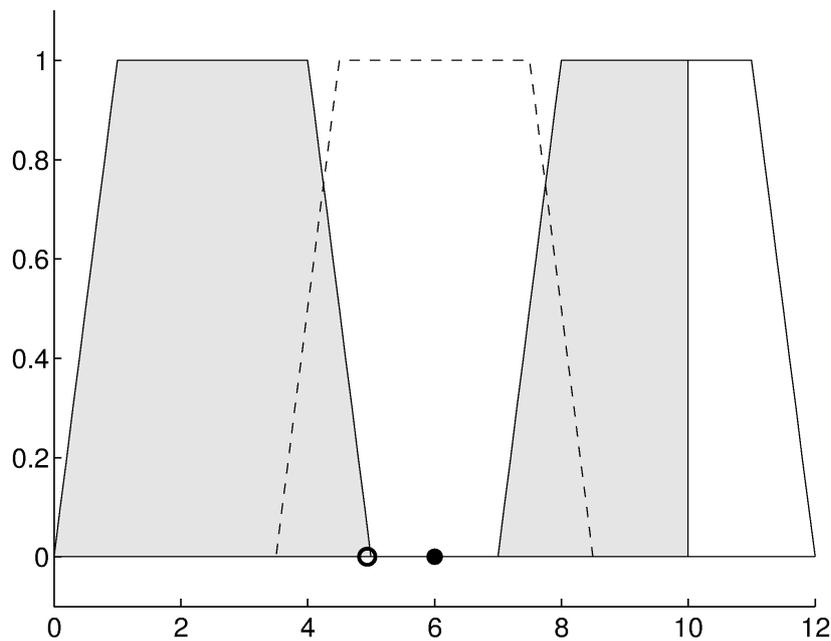


Figure 4.12: Controversial case

In case of classical Mamdani method, the output is not affected by the rightmost part of the aggregated function, as a result the output of the COG will shift to the left. The ABC method does not need the range parameter, therefore both trapezoids are equally considered during defuzzification.

4.5.2 Impact of the fuzziness

In Fig. (4.13), we show a typical Mamdani situation where membership functions with different level of fuzziness need to be defuzzified. The value resulted from the classical Mamdani method (o) is close to the center of the left triangular membership function, despite the right hand-sided triangle is full valid. The ABC method intuitively considers the validities of both membership functions, therefore the result (•) is closer to the right triangle. The aggregated output function is denoted by the gray area, while the central triangle (dashed line) is the linear combination of the two output membership functions used by the ABC method.

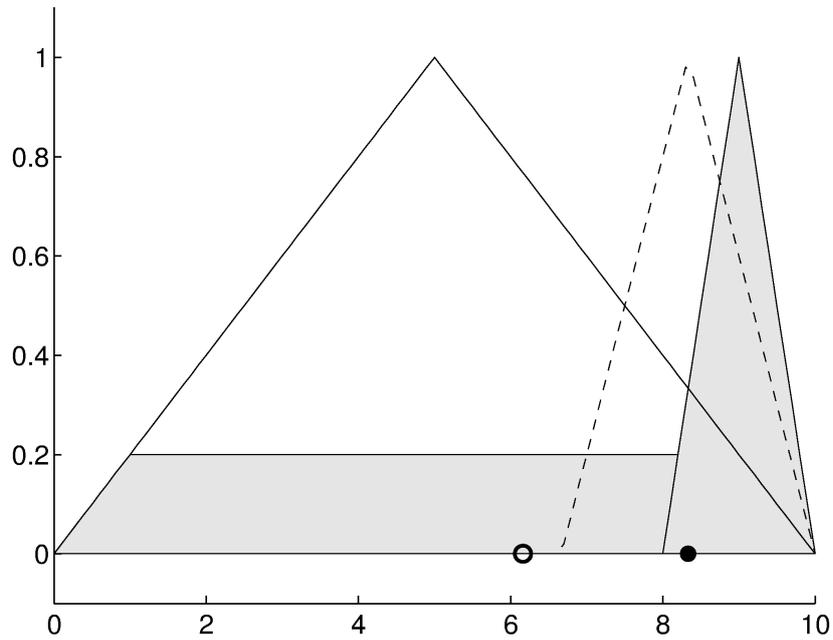


Figure 4.13: Impact of the fuzziness

4.5.3 Efficiency in Calculation

At classical Mamdani method, instead of the analytical integration, a numerical calculation is used. The classical and the ABC method were compared in Fig. (4.14) with different n numerical resolutions.

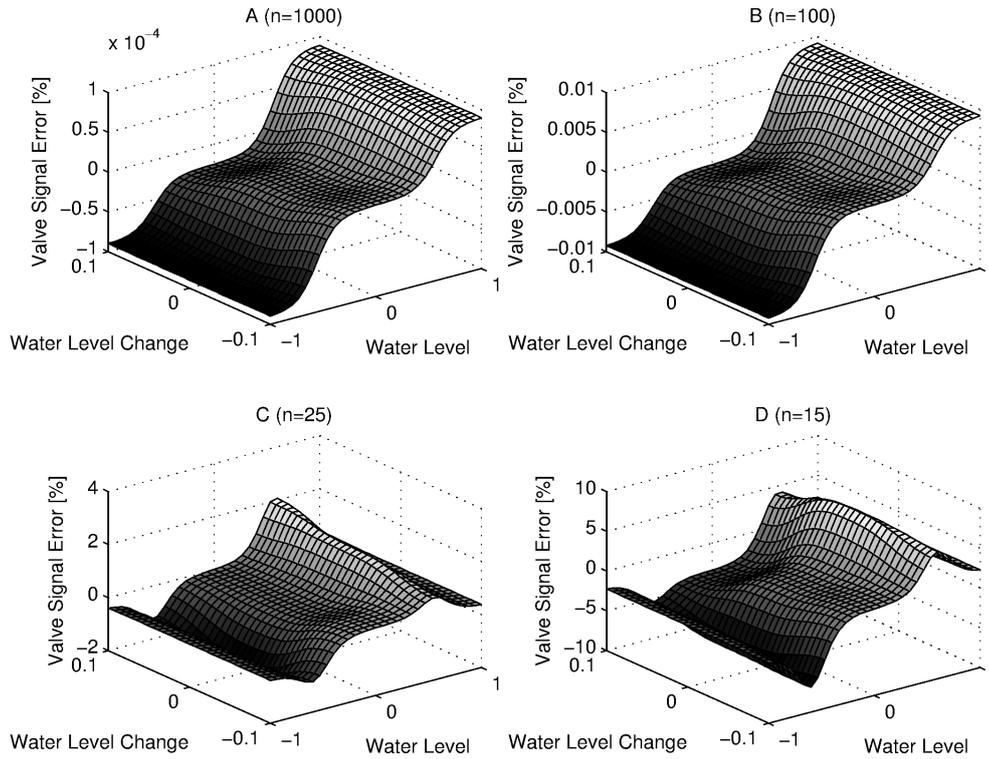


Figure 4.14: Membership function comparison

We can recognize that in our water tank example the error from analytical solution increases as the number of numerical base values decreases.

In Fig. (4.14A), $n = 1000$, $\epsilon \approx 10^{-4}$,

In Fig. (4.14B), $n = 100$, $\epsilon \approx 10^{-2}$,

In Fig. (4.14C), $n = 25$, $\epsilon \approx 2$,

In Fig. (4.14D), $n = 15$, $\epsilon \approx 8$.

It seems to be valid that if $n \rightarrow \infty$, we get the same result as with the ABC method. In this case, we can switch between Mamdani and ABC method using the same rule base.

4.5.4 Quality of the New Method

We tested the water tank demo control as described in Fig. (4.14). In Fig. (4.15) a typical test process can be seen in response to a rectangular reference signal. The dashed line shows the classical Mamdani response with $n = 15$ numerical resolution, while the solid line shows

the response of the presented ABC method with the same 5 rules. We can see the overshooting error is much smaller using the ABC method than with the classical Mamdani case. The speed of the calculation is still 2.5 times faster with the ABC method.

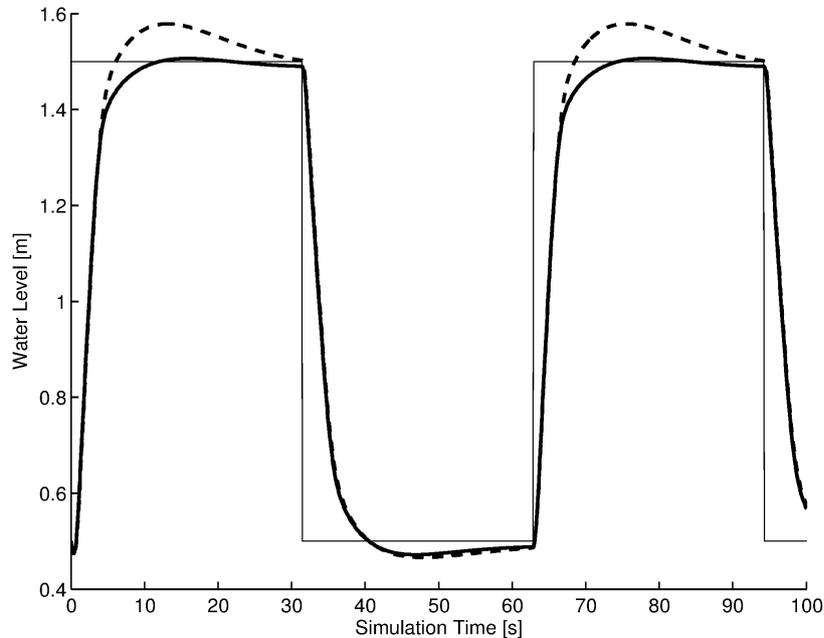


Figure 4.15: Response of the rectangular pulse with the classical Mamdani controller (- -) and ABC method (—)

4.5.5 Computational Speed

We have seen that the quality of the new approach is better, now we turn to the question of the efficiency. For the comparison, all values of the table are expressed in sec. All measurements were run 10^7 times in standard desktop environment. The algorithms were implemented in C. First, the classical Mamdani methods i.e. COG, BIS, SOM/LOM and MOM were compared. Their speed depends on the numerical resolution, and not on the number of rules. Since, the COG gives the best quality of calculation, it was chosen for the speed comparison. Table (4.1) shows the calculation speeds of implication (imp), aggregation (aggr) and COG defuzzification with different number of rules (3, 6, 9), and different numerical resolutions ($n = 101, n = 1001, n = 10001$). This latter influenced all of the measured operation speeds, while the COG did not depend on the number of applied rules.

CHAPTER 4. OPTIMIZED FUZZY CONTROLLER PROPERTIES AND APPLICATIONS

Table 4.1: Comparison of the classical defuzzification

	$n = 101$			$n = 1001$			$n = 10001$		
	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules
imp + aggr	0.154	0.326	0.481	1.688	3.852	5.885	24.268	47.711	66.952
COG	0.034	0.034	0.034	0.336	0.339	0.334	3.387	3.347	3.347
imp + aggr + COG	0.188	0.360	0.515	2.024	4.191	6.219	27.656	51.059	70.300

Two different defuzzification technique can be used for the ABC method:

1. The maximal probability (Mprob), see (4.12).
2. The maximal stability (Mstab), see (4.13).

In table (4.2), two approaches were compared with (3, 6, 9) number of rules. The calculation speed of Mprob and Mstab did not depend on the number of rules, similarly to the classical defuzzifications. However, both proposed methods vastly outperformed the classical COG.

Table 4.2: Comparison of the two new method

	3 rules	6 rules	9 rules
ABC Mprob	0.0008706	0.00086079	0.00086061
ABC Mstab	0.00078055	0.00077053	0.00077054

In table (4.3), the required computational time of the new ABC method versus classical Mamdani concept was summarized. The calculation speed of the whole ABC method depends only on the number of rules, as a result, the ABC method increasingly outperforms the classical solution as the numerical resolution increases, while it always gives more accurate result. If the resolution is low ($n = 101$), the speed is 15 times faster. If the resolution is medium ($n = 1001$), the speed is 190 times faster. If the resolution is high ($n = 10001$), the speed is 2400 times faster.

Table 4.3: The acceleration of the speed of the ABC method versus classical Mamdani concept

	$n = 101$			$n = 1001$			$n = 10001$		
	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules
imp + aggr + COG	0.188	0.360	0.515	2.024	4.191	6.219	27.660	51.059	70.300
ABC aggr + MStab	0.012	0.022	0.033	0.011	0.022	0.033	0.012	0.022	0.033
(imp + aggr + COG) / (ABC aggr + MStab)	15.960	16.543	15.716	187.801	192.212	189.665	2344.781	2343.707	2142.582

4.5.6 Optimizing the Fuzzy Structure

While the previous experiments were carried out by using the demo controller structure, it is possible to improve the control quality by redesigning the rules and membership functions. The experimental controllers in this section were optimized with a genetic algorithm using a fitness function that minimizes the sum of the error.

The water level control demo is originally controlled with five rules. Keeping the number of the rules but changing all the membership functions to sigmoid ones, and applying the Pliant inference system described in [9] reduced the error significantly as seen in Fig. (4.16).

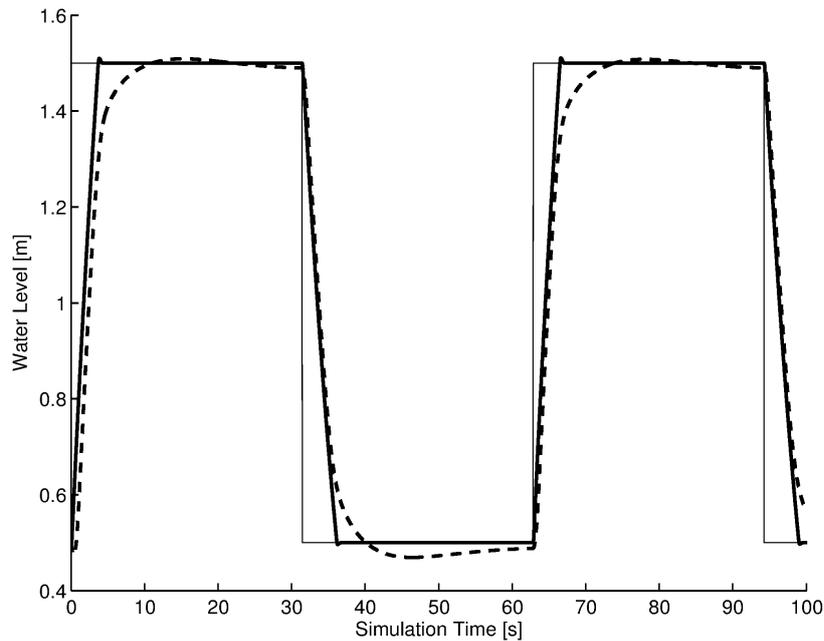


Figure 4.16: Water tank response of the rectangular pulse with the default Mamdani controller (- -) and the Pliant control (—)

The error of the optimized Pliant controller in this case was reduced by 25.4% with a barely noticeable overshoot.

It was also possible to further optimize the other classical fuzzy control demo, the inverted pendulum (also known as the cart and pole) problem. In this demo, the task is to position a cart with an unactuated pendulum on the top with a controlled force only applied to the cart. For fair comparison, the optimized controller was allowed to use the same level of peak force and sum of the applied unsigned force. The output of the simulation with the default and the optimized Pliant controller can be seen in Fig. (4.17).

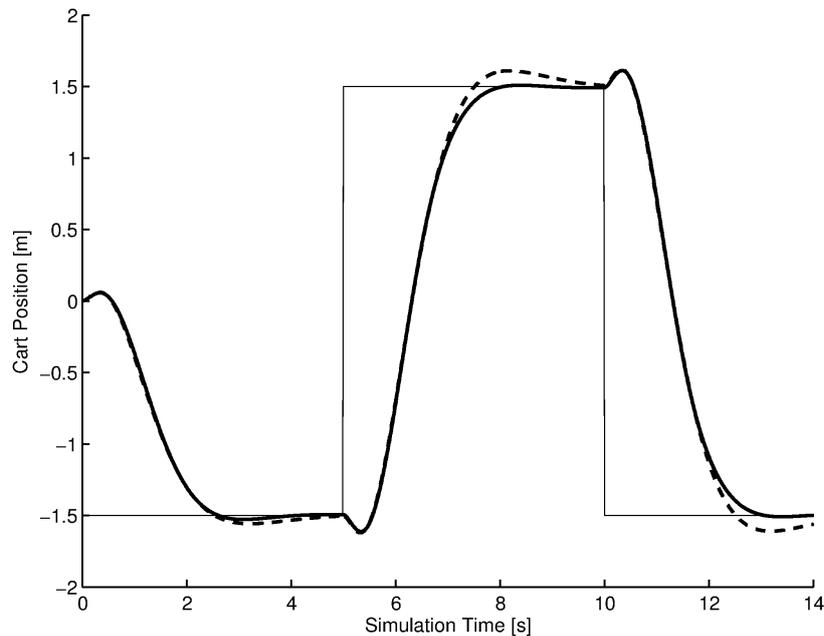


Figure 4.17: Inverted pendulum response of the rectangular pulse with the default Mamdani controller (- -) and the Pliant control (—)

In this case, the optimized controller could use only two rules to balance the pendulum on the cart, while the default controller applied 16 rules. While keeping all the constraints, the sum of absolute error was 1.47% lower with the optimized Pliant compared to the default controller during the simulation.

4.6 Benefits in the Revised Methods

In this work, we presented a new arithmetic based fuzzy control approach. This new method is based on simple arithmetic operations instead of implication which calculates the linear combination of the trapezoid membership functions. Furthermore, two new defuzzification operations were also introduced. This altogether forms a new fuzzy controller, which outperforms the classical approach in accuracy and speed. It is also range independent and intuitively handles different levels of fuzziness. Genetic algorithm based optimization can also be successfully applied to the proposed fuzzy controllers in order to boost the performance.

Acknowledgment

I would like to thank József Dombi for his guiding in his theoretical results.

I would like to thank Samy F. M. Assal for helping me in writing the article for the new results of the trajectory tracking approach.

I also thank Mária Kovács, Zoltán Gingl and all my colleagues for their support in writing the thesis and all related material.

Bibliography

- [1] Roberts, Adam. The History of Science Fiction. *Palgrave Histories of Literature, Palgrave Macmillan, New York*, pp. 168, 2006.
- [2] Shukla, Manshi, and Amar Nath Shukla. Growth of Robotics Industry Early in 21st Century. *International Journal of Computational Engineering Research*, Vol. 2, pp. 1554–1558, 2012.
- [3] IFR International Federation of Robotics website. <http://www.ifr.org/>
- [4] Poole, Harry H. Types of Robots. *Fundamentals of Robotics Engineering, Springer Netherlands*, pp. 27–51, 1989.
- [5] All On Robots website. <http://www.allonrobots.com/types-of-robots.html>
- [6] T. Szepe. Accurate force function approximation for pneumatic artificial muscles. *Logistics and Industrial Informatics (LINDI). 3rd IEEE International Symposium on IEEE, Budapest, Hungary*, 2011.
- [7] Tamás Szépe. Optimizing torque characteristic of antagonistic PAM actuators with a nonlinear torque transmission. *Advanced Robotics - publication under review*.
- [8] Tamas Szepe; Samy F.M. Assal. Pure Pursuit Trajectory Tracking Approach: Comparison and Experimental Validation. *International Journal of Robotics and Automation, Acta Press*, Vol. 27(4), pp. 355–363, 2012.
- [9] József Dombi, Tamás Szépe. Pliant control system: implementation. *Intelligent Systems and Informatics (SISY)*, 8, 2010.

BIBLIOGRAPHY

- [10] József Dombi, Tamás Szépe. Fuzzy Control Revisited. *PLOS ONE - publication under review*.
- [11] F. Daerden, D. Lefeber, B. Verrelst and R.V. Ham. Pleated pneumatic artificial muscles: actuators for automation and robotics. *In Proceedings of the IEEE/ASME International Conference on Advanced intelligent mechatronics, Como, Italy*, pp. 738–743, 2001.
- [12] B.R. Verrelst, R.V. Ham, B. Vanderborght, F. Daerden, D. Lefeber and J. Vermeulen. The pneumatic biped Lucy actuated with pleated pneumatic artificial muscles. *Autonomous Robots*, Vol. 18, pp. 201–213, 2005.
- [13] N. Saga, T. Nakamura and K. Yaegashi. Mathematical model of pneumatic artificial muscle reinforced by straight fibers. *J. Intell. Mater. Systems Structs*, Vol. 18, pp. 175–180, 2007.
- [14] C.P. Chou and B. Hannaford. Static and dynamic characteristics of McKibben pneumatic artificial muscles. *In Proceedings of the IEEE International Conference on Robotics and automation, San Diego, California, USA*, pp. 281–286, 1994.
- [15] C.P. Chou and B. Hannaford. Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE Trans. Robotics and Automn*, Vol. 12(1), pp. 90–102, 1996.
- [16] N. Tsagarakis and D.G. Caldwell. Improved modeling and assessment of pneumatic muscle actuators. *In Proceedings of the IEEE International Conference on Robotics and automation, San Francisco, California, USA*, pp. 3641–3646, 2000.
- [17] G.K. Klute and B. Hannaford. Accounting for elastic energy storage in McKibben artificial muscle actuators. *Trans. ASME, J. Dynamic Systems, Measmt, Control*, Vol. 22(2), pp. 386–388, 2000.
- [18] S. Davis, N. Tsagarakis, J. Canderle and D.G. Caldwell. Enhanced modeling and performance in braided pneumatic muscle actuators. *Int. J. Robotics Res.*, Vol. 22(3-4), pp. 213–217, 2003.

BIBLIOGRAPHY

- [19] A. Sanchez, V. Mahout and B. Tondu. Nonlinear parametric identification of a McKibben artificial pneumatic muscle using flatness property of the system *In Proceedings of the IEEE International Conference on Control application, Trieste, Italy*, pp.70–74, 1998.
- [20] F. Daerden, D. Lefeber. Pneumatic Artificial Muscles: actuators for robotics and automation. <http://www.docstoc.com/docs/68837564/Pneumatic-Artificial-Muscles-actuators-for-robotics-and-automation>
- [21] FESTO specification http://www.festo.com/cms/en-us_us/5030.htm
- [22] FESTO specification http://www3.festo.com/_C1256D56002E7B89.nsf
- [23] T. Szepe, J. Sarosi. Model Based Open Looped Position Control of PAM Actuator *IEEE 8th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia*, 2010.
- [24] P. Toman, J. Gyeviski, T. Endrody, J. Sarosi and A. Veba. Design and Fabrication of a Test-bed Aimed for Experiment with Pneumatic Artificial Muscle *International Journal of Engineering, Annals of Faculty of Engineering Hunedoara, ISSN 1584-2665*, pp. 91-94., Vol. 7, No. 4, pp. 91–94, 2009.
- [25] T. Szepe, J. Sarosi. Matlab Models for Pneumatic Artificial Muscles. *Scientific Bulletin of the "Politehnica", Timisoara, Romania*, 2009.
- [26] B. Tondu, P. Lopez. Modelling and control of McKibben artificial muscle robot actuator. *IEEE Control System Magazine*, Vol. 20, pp- 15–38, 2000.
- [27] T. Kerscher, J. Albiez, J.M. Zăşllner and R. Dillmann. FLUMUT - Dynamic Modelling of Fluidic Muscles using Quick-Release. *3rd International Symposium on Adaptive Motion in Animals and Machines, Ilmenau, Germany*, 2005.

BIBLIOGRAPHY

- [28] A. Hosovsky, M. Balara. Pneumatic Artificial muscle force function approximation using ANFIS. *Journal of applied science in the thermodynamics and fluid mechanics*, ISSN 1802-9388, Vol. 3, No. 1, 2009.
- [29] D. Feil-Seifer, M.J. MatariĀŚ. Human Robot Interaction. *Encyclopedia of Complexity and Systems Science*, 2009.
- [30] ABB website. <http://new.abb.com/products/robotics/yumi>
- [31] KUKA website. http://www.kukaconnect.com/wp-content/uploads/2012/07/KUKA_LBR4plus_ENLISCH.pdf
- [32] Fanuc website. [http://fanuc.co.jp/en/product/catalog/pdf/robot/RCR-35iA\(E\)_01.pdf](http://fanuc.co.jp/en/product/catalog/pdf/robot/RCR-35iA(E)_01.pdf)
- [33] Rethink Robotics website. <http://www.rethinkrobotics.com/build-a-bot/baxter/>
- [34] Universal Robots website. <http://www.universal-robots.com/products/ur3-robot/>
- [35] A. Parmiggiani, G. Metta, N. Tsagarakis. The mechatronic design of the new legs of the iCub robot. *Humanoid Robots (Humanoids), 12th IEEE-RAS International Conference*, 2012.
- [36] C. Semini. HyQ–Design and development of a hydraulically actuated quadruped robot. *PD Thesis, Genoa: University of Genoa*, 2010.
- [37] A. Kapelevich High Gear Ratio Epicyclic Drives Analysis. *Gear Technology*, pp. 61–67, 2014.
- [38] J.W. Sensinger, J.H. Lipsey. Cycloid vs. harmonic drives for use in high ratio, single stage robotic transmissions. *Proceedings-IEEE International Conference on Robotics and Automation, Minnesota, USA*, 2012.
- [39] T. Hesselroth, K. Sarkar, P.P. van der Smagt, et al. Neural network control of a pneumatic robot arm. *Systems, Man and Cybernetics, IEEE Transactions*, Vol. 24(1), pp. 28–38, 1994.

BIBLIOGRAPHY

- [40] J.F. Zhang, C.J. Yanga, Y. Chena et al. Modelling and control of a curved pneumatic muscle actuator for wearable elbow exoskeleton. *Mechatronics*, Vol. 18, pp. 448–457, 2008.
- [41] N. Schmit, O. Masafumi. Design and realization of a non-circular cable spool to synthesize a nonlinear rotational spring. *Advanced Robotics*, Vol. 26, pp. 234–251, 2012.
- [42] G. Campion, G. Bastin, and B. D. Novel, Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans. on Robot. and Autom.*, 12(1), pp. 47–62, 1996.
- [43] L. Gracia, and J. Tornero, Kinematic control of wheeled mobile robots. *Latin American Applied Research.*, 38, pp. 7–16, 2008.
- [44] C. Louste, and A. Liegeois, Path planning for non-holonomic vehicles: a potential viscous fluid field method. *Robotica*, Vol. 20, pp. 291–298, 2002.
- [45] J. Borenstein, and Y. Koren, The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Trans. on Robot. and Autom.*, Vol. 7, pp. 278–288, 1991.
- [46] D. Fox, W. Burgard, and S. Thrun, The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazin*, Vol. 4, pp. 23–33, 1997.
- [47] A. Stentz, and I. C. Mellon, Mellon, Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation*, Vol. 10, pp. 89–100, 1995.
- [48] S. Aydin, and H. Temeltas, Fuzzy-differential evolution algorithm for planning time-optimal trajectories of a unicycle mobile robot on a predefined path. *Advanced Robotics*, Vol. 18, pp. 725–748, 2004.
- [49] I. Waheed, and R. Fotouhi, Trajectory and temporal planning of a wheeled mobile robot on an uneven surface. *Robotica*, Vol. 27, pp. 481–498, 2009.

BIBLIOGRAPHY

- [50] J. Thomas, A. Blair, and N. Barnes, Towards an efficient optimal trajectory planner for multiple mobile robots. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vol. 3, pp. 2291–2296, 2003.
- [51] R.W. Brockett, R.S. Millman, H.J. Sussmann, Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, Birkhauser, Boston, MA, pp. 181–191, 1983.
- [52] C. Samson, Time-varying feedback stabilization of car-like wheeled mobile robots. *International Journal of Robotics Research*, Vol. 12, pp. 55–64, 1993.
- [53] C. Canudas de Wit, Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Trans. on Automat. Contr.*, Vol. 37, pp. 1791–1797, 1992.
- [54] P. Morin, and C. Samson, Practical stabilization of a class of nonlinear systems: application to chain systems and mobile robots. *IEEE Conf. Decision Control*, Vol. 39, pp. 2144–2150, 2000.
- [55] G. Oriolo, A. De Luca, and M. Vandittelli, WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, Vol. 10, pp. 835–852, 2002.
- [56] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, A stable tracking control method for an autonomous mobile robot. *Proc. IEEE Int. Conf. on Robot. and Auto*, pp. 384–389, 1990.
- [57] A. De Luca, G. Oriolo, and M. Vendittelli, Control of wheeled mobile robots: an experimental overview. *RAMSETE-Articulated and Mobile Robotics for Services and Technologies*, Springer-Verla, 2001.
- [58] A. De Luca, and M. D. D. Benedetto, Control of nonholonomic systems via dynamic compensation. *Kybernetika*, Vol. 29, pp. 539–808, 1993.

BIBLIOGRAPHY

- [59] F. Kuhne, W.F. Lages, and J.M. G. Silva Jr, Model predictive control of a mobile robot using linearization. *Proc. of Mech. and Robot*, pp. 525–530, 2004.
- [60] O. Amidi, Integrated Mobile Robot Control. *Rep. CMURI-TR-90-17, Robotics Institute, Carnegie Mellon University, Pittsburgh, USA*, 1990.
- [61] A. Ollero, A. Garc a-Cerezo, and J. Mart nez, Model Fuzzy supervisory path tracking of mobile robots. *Control Engineering Practice*, Vol. 2, pp. 313–319, 1994.
- [62] S. Singh, and D. H. Shin, Position based path tracking for wheeled mobile robots. *Proc. of IEEE/RSJ Int. Workshop on Intelligent Robot Systems*, pp. 386–391, 1989.
- [63] A. Ollero, and G. Heredia, Stability analysis of mobile robot path tracking. *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vol. 3 pp. 461–466, 1995.
- [64] F. Golnaraghi and B. C. Kuo, Automatic Control Systems. *9th edition, Wiley, ISBN 978-0-470-04896-2*, 2009.
- [65] W. Yu, O. Chuy, E. G. Collins, and P. Hollis, Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle. *IEEE Transactions on Robotics*, pp. 440–453, 2010.
- [66] S. Ahmed, M. N. Karsiti, and R. N. K. Loh, Multiagent Systems. *Chapter 5, ISBN 978-3-902613-51-6*, pp. 103-127, 2009.
- [67] M. Brezak, I. Petrovic, and N. Peric, Experimental comparison of trajectory tracking algorithms for nonholonomic mobile robots. *Proc. of IEEE Int. Conf. on Industrial Electronics*, pp. 2229–2234, 2009.
- [68] R. C. Coulter, Implementation of the pure pursuit path tracking algorithm. *Tech. Rep. CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pa, USA*, 1992.
- [69] Koala robot specifications. <http://www.k-team.com/mobile-robotics-products/koala/specifications>

BIBLIOGRAPHY

- [70] J. Borenstein and L. Freng, Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, Vol. 12, pp. 869–880, 1996.
- [71] S. Assilian. *Artificial intelligence in the control of real dynamical systems*. Ph.D. Thesis, London University, Great Britain, 1974.
- [72] E. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. IEE*, Vol. 121, pp. 1585–1588, 1974.
- [73] D. Filev and R. Yager. A generalized defuzzification method via BAD distributions. *Internat. J. Intell. Systems*, Vol. 6, pp. 689–697, 1991.
- [74] R. Yager and D. Filev. SLIDE: a simple adaptive defuzzification method. *IEEE Trans. Fuzzy Systems*, Vol. 1, pp. 69–78, 1993.
- [75] T. Jiang and Y. Li. Generalized defuzzification strategies and their parameter learning procedures. *IEEE Trans. Fuzzy Systems*, Vol. 4, pp. 64–71, 1996.
- [76] S. Roychowdhury and B.-H. Wang. Cooperative neighbors in defuzzification. *Fuzzy Sets and Systems*, Vol. 78, pp. 37–49, 1996.
- [77] Q. Song and R. Leland. Adaptive learning defuzzification techniques and applications. *Fuzzy Sets and Systems*, Vol. 81, pp. 321–329, 1996.
- [78] S. Roychowdhury and W. Pedrycz. A survey of defuzzification strategies. *Internat. J. Intell. Systems*, Vol. 16, pp. 679–695, 2001.
- [79] W. Van Leekwijck and E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, Vol. 108, pp. 159–178, 1999.
- [80] E. Van Broekhoven and B. De Baets. Fast and accurate centre of gravity defuzzification of fuzzy system outputs defined on trapezoidal fuzzy partitions. *Fuzzy Sets and Systems*, Vol. 157, pp. 904–918, 2006.

BIBLIOGRAPHY

- [81] D. Driankov, H. Hellendoorn and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer, Berlin, 1993.
- [82] A. Patel and B. Mohan. Some numerical aspects of center of area defuzzification method. *Fuzzy Sets and Systems*, Vol. 132, pp. 401–409, 2002.
- [83] A. Sakly and M. Benrejeb. *Activation of trapezoidal fuzzy subsets with different inference methods*. In: T. BilgiÄŒ, B. De Baets, O. Kaynak (Eds.), *Fuzzy Sets and Systems - Proc. IFSA 2003, Lecture Notes in Artificial Intelligence*, Vol. 2715, Springer, Berlin, 2003, pp. 450–457.
- [84] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, Part I:8, 199–249, Part II:8, 301–357, Part III:9, 43–80, 1975.
- [85] R. Jain. Tolerance analysis using fuzzy sets. *Int. J. System Science*, 7, pp. 1393–1401, 1976.
- [86] M. Mizumoto and K. Tanaka. The four operations of arithmetic on fuzzy numbers. *Syst. Comput. Controls*, 7(5), pp. 73–81, 1976.
- [87] M. Mizumoto and K. Tanaka. Algebraic properties of fuzzy numbers. *Proc. Int. Conf. On Cybernetics and Society*, Washington, DC, pp. 559–563, 1976.
- [88] S. Nahmias. Fuzzy variables. *Fuzzy Sets and System*, 1, pp. 97–110, 1978.
- [89] H. T. Nguyen. A note on the extension principle for fuzzy sets. *J. Math. Anal. Appl.*, 64, pp. 369–380, 1978.
- [90] D. Dubois and H. Prade. Operations on fuzzy numbers. *Int. J. Systems Science*, 9, pp. 613–626, 1978.
- [91] R. C. Young. The algebra of many-valued quantities. *Math. Ann.*, 104, pp. 260–290, 1931.

BIBLIOGRAPHY

- [92] D. Dubois and H. Prade. Fuzzy members: An overview. *Analysis of Fuzzy Information*, Vol. I., CRC Press, Boca Raton, FL, pp. 3–39, 1987.
- [93] A. Kaufmann and M. M. Gupta. *Introduction to Fuzzy Arithmetic- Theory and Applications*. Van Nostrand Reinhold, New York, 1985.
- [94] A. Kaufmann and M. M. Gupta. *Fuzzy Mathematical Models in Engineering and Management Science*. North-Holland, Amsterdam, 1988.
- [95] M. Mares. *Computation Over Fuzzy Quantities*. CRC Press, Boca Raton, FL, 1994.
- [96] D. Dubois and H. Prade. Special Issue on Fuzzy Numbers. *Fuzzy Sets and System*, 24(3), 1987.
- [97] R. Fullér and R. Mesiar. Special Issue on Fuzzy Arithmetic. *Fuzzy Sets and System*, 91(2), 1997
- [98] Dombi, József. Pliant Arithmetics and Pliant Arithmetic Operations. *Acta Polytechnica Hungarica*, 6.5, pp. 19–49, 2009.

SUMMARY

There are different types of classifications for robotic devices. For example, these can be categorized by the degrees of freedom, the implemented level of intelligence, and the application or by the way of motion. Furthermore, there is a newly emerged field, the so-called collaborative robotics, where the human safety approach, known originally from the domestic robots, is introduced to the operation of the industrial manipulators.

Basic researches aim towards the application of pneumatic artificial muscles (PAMs), since their inherently elastic, these relatively new type of actuators are optimal for the development of collaborative robots. Furthermore, PAM actuators can be built into robotic joints as well, and the manufactured joint will be also flexible because of their inherently elasticity. However, controlling of the PAMs is a difficult problem due to the highly nonlinear characteristics of the generated force.

In this study, I developed a new model using six parameters which accurately models the static characteristics of PAM actuators. This transfer function expresses the generated force as a function of the contraction and the pressure. The results showed that, the compactness and the precision were further improved compared to other compact models. Furthermore, this function is especially useful in the usual working range of the PAMs, because of its lower error rate, and the reduced number of parameters allows faster and more reliable model fitting.

The pressure required to maintain a given contraction with a given force, which is necessary for open-looped applications, can be expressed by using the inverse of the transfer function. A recalibration method was also proposed, in which only a limited number of

measurements are required to refit the model with remarkable precision.

I also developed a simulation framework capable of optimizing nonlinear transmission systems. Furthermore, a new non-circular shaped pulley was also designed which improved the efficiency of the force to torque transmission in an antagonistic setup of PAM actuators. I also designed an experimental transmission unit to validate the capabilities of the theoretical results, which showed significant improvements. I measured the gear reduction capabilities of the physical model as well and my results showed high correlation with the simulated model behavior.

The precise trajectory tracking is another important aspect in the field of robotics which is important both for the mobile and legged robots. Therefore, I developed a new concept for trajectory tracking of differential-drive mobile robots, in which an updating time dependent lookahead distance related to the sampling time was introduced. The developed method also avoids the selection of the tuning and damping parameters used by the well-known linear and nonlinear state tracking controllers. In the contrary, this new controller requires only one application and device independent parameter.

This concept was also implemented to a mobile robot. Furthermore, this new model was compared with the commonly used linear and nonlinear state tracking controllers. The results showed significant tracking error reductions and a high reliability with the new optimized trajectory tracking controller.

Finally, application of a fuzzy control system might be a good choice to develop an efficient robotic device or to operate the actuators effectively and it could be also useful for higher controlling levels.

Using the fuzzy arithmetic theories of József Dombi, I developed two new fuzzy controllers which can be used in practical MATLAB simulations as well. I also compared them with the classical Mamdani fuzzy control. All versions rule fitting technique remained the same as originally defined, while the implication, aggregation, and defuzzification methods were modified for the Pliant and Arithmetic Based Control (ABC) cases. The classical fuzzy controller utilizes the well-known implication and aggregation methods based on the numer-

SUMMARY

ical representation of the output membership functions and the classical defuzzification was the center of gravity (COG) method. I implemented for the first time, the frameworks for the whole Pliant and ABC fuzzy control in MATLAB environment. Furthermore, I also implemented new fuzzy structures in these frameworks for controlling typical demo systems, and compared their performance to the classical fuzzy controller.

My results showed that the implemented systems outperform the classical fuzzy controller in accuracy and speed. The new methods are also range independent and can also handle intuitively the different levels of fuzziness. I also applied genetic algorithm based optimization to the proposed fuzzy controller in order to boost the performance.

ÖSSZEFOGLALÓ

A robotikai eszközök sokfélesége miatt csoportosításuk is változatos lehet. Kategorizálásuk történhet pl. a szabadsági fokok száma, az applikáció és a mozgás típusa alapján is. Azonban az utóbbi évek során egy újabb irányzat kezd kibontakozni, a kollaboratív robotika. Ennél az irányzatnál a háztartási robotoknál jól ismert felhasználói biztonságot igyekeznek az ipari manipulátorokra is alkalmazni. Ennek köszönhetően az alapkutatások egy jelentős része a pneumatikus műizmok felé fordult (PAM), mert szerkezeti rugalmasságuknak köszönhetően, ezek az aktuátorok kiválóan alkalmasak pl. kollaboratív robotikai alkalmazásokhoz. Például a PAM-ek robot csuklóba is beépíthetőek, így rugalmasságukból kifolyólag az elkészült manipulátor is ezen tulajdonsággal rendelkezik. Az alkalmazhatóságot azonban megnehezíti a PAM-ek szabályozása, mivel az általuk kifejtett erő nem lineáris karakterisztikát mutat.

Ebben a tanulmányban egy hat paramétert használó új modellt hoztam létre, amely alkalmas a PAM aktuátorok statikus karakterisztikájának leírására. Ez az összefüggés a PAM által létrehozott erőt képes kifejezni az összehúzódás és az alkalmazott nyomás függvényében. Más modellekkel összehasonlítva, az új függvény pontosabb és alakja kompaktabb. Ezen kívül a PAM-ek normál működési tartományán belül különösen jól hasznosítható, mivel ezen a tartományon belül nagyon alacsony a modell hibája. A kompakt modell további előnye, hogy más modellekhez képest a kevesebb paraméter használata gyorsabb és megbízhatóbb modell illesztést tesz lehetővé.

Ebben a munkában egy olyan szimulációs keretrendszert is kifejlesztettem, amely a nem lineáris karakterisztikát mutató transzmissziós rendszerek optimalizálására alkalmas. Ennek segítségével egy excentrikus, csiga alakú áttételt terveztem, amely jelentősen javítja az erő-

nyomaték átvitelt az antagonisztikus elrendezésű PAM aktuátorok esetében. Ezen kívül egy kísérletes eszközt is készítettem, amely tartalmazta a csiga áttételt is. Ennek segítségével a szimulációs eredményeket sikerült visszaigazolni, mely jelentős javulást mutatott az erőnyomaték átalakítása során.

Egy másik fő alapkutatósi irány a megfelelő pontosságú trajektória követés megvalósítása, mely egyaránt fontos mind kerekeken guruló, mind lépegető robotok esetében. Kifejlesztettem egy új eljárást, mely differenciál hajtású kerekeken gördülő mobil robotok trajektória követését teszi lehetővé. Ebben olyan előretekintési távolságot határoztam meg, mely az idő függvényében folyamatosan frissül. Továbbá a hagyományos trajektóriakövető megoldásoknál szükséges tuningoló és csillapító paraméterekre ennél a módszernél nincsen szükség. Helyettük a "pure pursuit" trajektória követő eljáráshoz csak egyetlen, alkalmazás és eszköz független paramétert kell meghatározni.

A kísérleteket a differenciál meghajtású Koala II roboton végeztem el. Mind az előrecsatolt, mind az állapotkövető eljárások, mind pedig a "pure pursuit" trajektóriakövető eljárásokat többszöri futtatásuk alapján hasonlítottam össze. Eredményül az újonnan kifejlesztett eljárás közel ötödére csökkentette a pozíció és az orientációban mérhető hibákat a teszt trajektória esetében a két állapotkövető eljáráshoz képest.

A fuzzy szabályozás alkalmazásával elérhető, hogy hatékonyan működő robotot, vagy aktuátort fejleszthessünk, de magasabb szabályozási szintek megvalósítására is alkalmas lehet.

Dombi József elméletén alapuló fuzzy aritmetikán, ahol a műveleteket a függvények inverzén végezzük el, lehetőség van számításilag hatékony módon megoldani a teljes Mamdani típusú fuzzy szabályozást. Amennyiben háromszög vagy trapéz jellegű tagsági függvényeket alkalmazunk, úgy a bemutatott ABC, illetve szigmoid jellegű tagsági függvények esetében Pliant szabályozót kapunk. Mindkét változat szabályillesztési megoldása a klasszikus megvalósítással készült, de az implikáció, aggregáció és a defuzzifikáció megvalósítása az új koncepciót követi. A klasszikus fuzzy szabályozásnál az implikáció és aggregáció műveletek megvalósítása a kimeneti tagsági függvények numerikus reprezentálásából adódóan szín-

tén numerikus módszerrel történik, míg a klasszikus defuzzifikálás művelete a súlyközéppont módszere. Elsőként implementáltam keretrendszereket a Pliant és ABC fuzzy szabályozókhoz MATLAB környezetben, melyekkel tipikus fuzzy demó szabályozásokat oldottam meg. A szimulációs eredményeket a klasszikus fuzzy szabályozásokkal összevetve jelentős javulást mutattam ki.

Kimutattam, hogy a saját keretrendszerben futtatott szabályozók mind sebességben, mind pedig pontosságban jobb eredményeket adnak a klasszikus fuzzy szabályozáshoz képest. Kimutattam a tartományfüggetlenségüket, és a különböző fuzziáságú tagsági függvények intuitív kezelésének képességüket, melyek a gyakorlati alkalmazásban nyújtanak előnyt.

Sikerrel alkalmaztam genetikus algoritmust, amivel az új keretrendszerbe implementált fuzzy szabályozások szabályozó képességét javítottam fel.