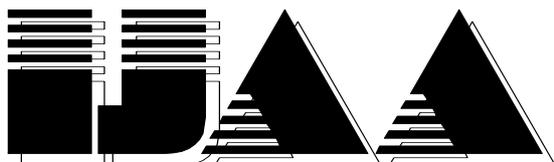


INTERNATIONAL JOURNAL AUTOMATION AUSTRIA



SPECIAL ISSUE

2005

Contents

Pages

Preface

ROBOTS

<i>ADVANCES IN ROBOTICS</i>	1
Peter Kopacek (AUT)	
<i>HIGH SPEED ROBOT VISION AUTHENTICATION</i>	13
Wael Adi (UAE), Ali Dawood , Ali Mabrouk (GER)	
<i>CONTINUOUS GAIN SCHEDULING FOR UNDERACTUATED SYSTEMS</i>	25
Y.Tachwali, M.A.Jarrah (UAE)	

ROBOT SOCCER

<i>THE FUZZY INFERENCE SYSTEM FOR HUMANOID ROBOT BALANCE CONTROL</i>	39
Shu-Yun Chung, Han-Pang Huang, Shu-Wen Yu (TPE)	
<i>CONTROL DESIGN FOR DIFFERENTIAL STEERING ROBOT</i>	51
Gregor Klancar, Drago Matko (SLO)	
<i>EVALUATION AND COMPARISON STUDY OF SHOOTING BALL ALGORITHMS</i>	63
Alsalem F M, and Al-Jarrah M (UAE)	

IDENTIFICATION AND MODELLING

<i>SYSTEM IDENTIFICATION USING A MODIFIED PARTIAL RECURRENT NETWORK</i>	73
Muaiad Shawqi AL-Faysale (IRQ)	
<i>COMPUTER AIDED MODELLING OF A MECHATRONIC SYSTEM</i>	85
M. Ebrahimi, A. A. Abdul-Ameer and R. Whalley (GBR)	
<i>MODELING OF THERMAL ANALYSIS OF ORBITAL SPACECRAFT-MULTILAYER</i>	101
<i>INSULATION SUBJECTED TO HYPERVELOCITY IMPACT DAMAGE</i>	
Mohamed A. Gadalla (UAE)	

VARIOUS

<i>A SELF ORGANIZED HOLONIC CONTROL FOR MECHATRONICS</i>	115
<i>COMPLEX SYSTEMS: APPLICATION TO A ROBOTIZED CAR PARK</i>	
Patrick Pujo, Fouzia Ounnar, Cecilia Zanni (FRA)	
<i>HARDWARE IN THE LOOP SIMULATOR FOR AUS-UAV SYSTEM</i>	127
Hasan M M, and Al-Jarrah M (UAE)	
<i>FREE FORMING OF MICROSTRUCTURES USING SYNCHRONOUS LOCALIZED</i>	139
<i>ELECTRO-DEPOSITION</i>	
R.A. Said (UAE)	

Author's Index	151
-----------------------------	-----

ADVANCES IN ROBOTICS

Peter Kopacek

Intelligent Handling and Robotics - IHRT
Vienna University of Technology, Austria
Favoritenstr. 9-11/3256, A-1040 Vienna, Austria
kopacek@ihrt.tuwien.ac.at

Abstract

The field of robotics is one of the most innovative in the last decade. We are moving now from conventional, unintelligent industrial robots to mobile, intelligent, cooperative robots. This new generation of robots offers a lot of new application fields. Some of them will be growing dramatically in the nearest future.

Therefore in this paper the present state will be discussed, selected applications described and an outlook on future developments will be given.

Keywords: Mechatronics, industrial robots, intelligent robots, cooperative robots.

1. Introduction

Conventional industrial robots from the late 70's are now only a tool on the production level. One of the oldest dreams of the robotic community – intelligent, mobile, cooperative as well as humanoid robots – starts to become reality not only because of the rapid development of “external” sensors and micro- and nanotechnology.

External sensors (e.g. visual, auditive, force-torque...) combined with micro drives, embedded systems,... offer intelligent robots the possibility to see, hear, speak, feel, smell like humans. Compared with conventional, unintelligent, industrial robots, intelligent robots fulfill new, innovative tasks in new application areas.

There are three “starting” points for the development of intelligent robots: Conventional, stationary industrial robots; mobile, unintelligent platforms (robots) and walking machines. Stationary industrial robots equipped with external sensors are used today for assembly and disassembly operations, fuelling cars, cleaning of buildings and airplanes, humanitarian demining ... and have been the first “intelligent” robots.

Mobile platforms with external sensors are available since some years and cover a broad application field. The core of each robot is an intelligent mobile platform with an on-board PC. On this platform, various devices, like arms, grippers, transportation equipment, etc., can be attached. Communication between the „onboard PC“ and the „supervisory PC“ is carried out by radio-based networks - communication with the environment can be accomplished by voice, beep or bell.

Walking machines or mechanisms are well known since some decades. Usually they have 4 to 6 legs (multiped) and only in some cases 2 legs (biped) – walking on two legs is from the

view of control engineering a very complex (nonlinear) stability problem. Biped walking machines equipped with external sensors are the basis for “humanoid” robots.

It was an old dream to have a personal robot looking like a human. Main features of a humanoid robot are

- biped walking
- voice communication – speech recognition
- facial communication

The main feature of a real human is the two legged movement and the two legged way of walking. Much research has been conducted on biped walking robots because of their greater potential mobility. On the other side they are relatively unstable and difficult to control in terms of posture and motion.

In addition these intelligent robots – especially mobile platforms and humanoid robots - are able to work together on a common task in a cooperative way. The goals are so called “Multi Agent Systems – MAS”. MAS consist of a distinct number of robots (agents), equipped with different arms, lifts, tools, gripping devices, ... and a host computer. MAS have to carry out a whole task e.g. assemble a car. The host computer divides the whole task in a number of different subtasks (e.g. assembling of wheels, windows, brakes, ...) as long as all this subtasks can be carried out by at least one agent. The agents will fulfill their subtasks in a cooperative way until the whole task is solved.

In industry intelligent robots will work together with humans in a cooperative way on a common working place.

2. Application examples

In the following some examples for introducing AI methods on a very low and cheap level in SME’s will be shortly described and discussed. These solutions were developed with and are realised in companies.

2.1. Disassembly cell for printed circuit boards

The layout of the cell is shown in Fig. 1. The basis of a disassembly cell is a very stiff frame construction developed from commercially available profiles. In a manual feeding station the PCBs with a maximum size of 300 x 220 mm are attached on special work holding device. The disassembly cell consists of 4 stations:

- Vision system
- Laser desoldering system
- Removal station
- Heating removal station

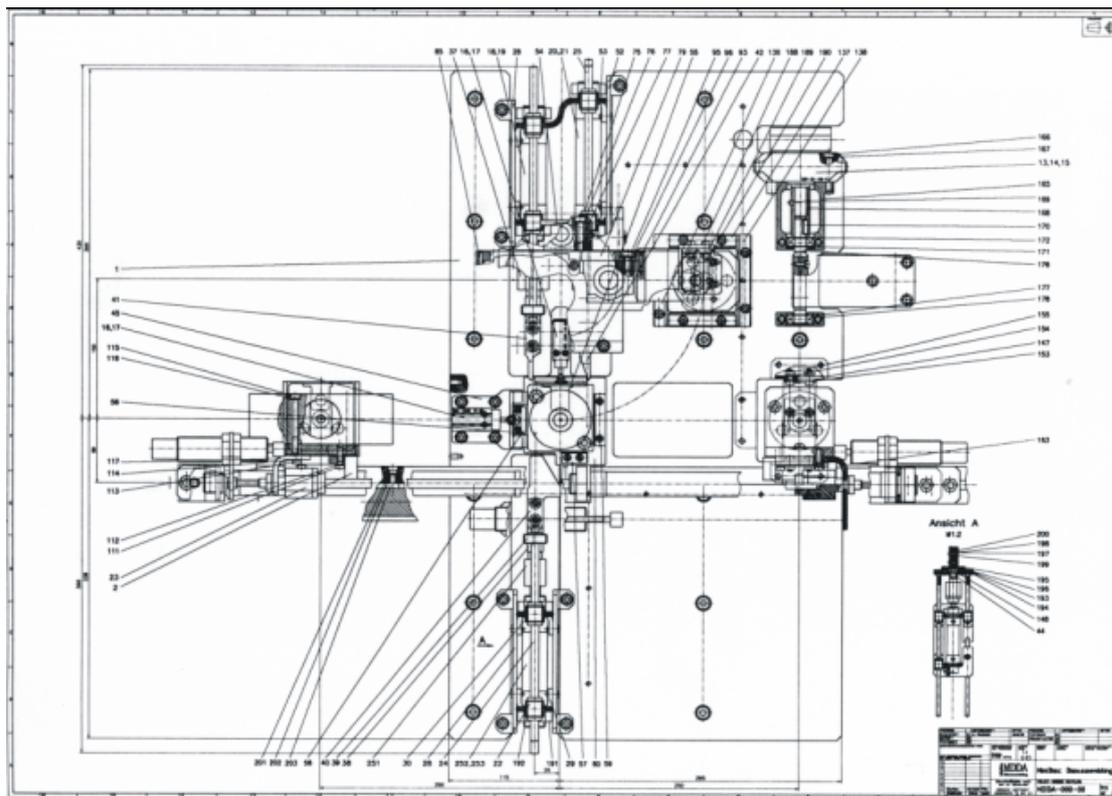


Fig. 1 Layout of the disassembly cell for PCB's (Kopacek, Noe, 2001)

The vision system has several tasks. It has to recognize the re-useable parts by means of a data base containing the data (kind, production company, assigned, dimensions). The vision system has to detect the re-useable part and to determine the position, the size and the centre of inertia. Furthermore it has to classify the useable parts to be desoldered or removed from sockets.

The desoldering station consists of a cross table – two linear axes – controlled to reach every point (centre of inertia) on the PCB. The desoldering process is carried out by laser technology. The desoldered parts are put on a distinct area outside the laser from which they are removed by the industrial robot and to put into the appropriate magazines.

The third station is the removal station for socket parts. An industrial robot equipped with special grippers as well as external sensors carries out process. The robot removes these parts and puts them also in the right magazines. A prototype of this disassembly cell is now in use since 3 years.

2.2. A semiautomatized Disassembly Cell for Mobile Phones (Kopacek, Kopacek, 2003)

After a detailed analysis of used mobile phones concerning the parts as well as the assembly technology and tests for disassembly with the most frequent mobile phones the following concept for the disassembly cell was created (Fig. 2). It consists of five automated stations plus a manual feeding and removal station:

- Feeding and removal station
- Drilling and milling station

- Removal station for the covers
- Drilling station
- Circuit board removal station
- Drilling station

For disassembly of the mobile phones they were fixed on a pallet in a distinct position. These pallets are moving around on a transportation system. According to the necessary disassembly operations the pallets with the mobile phones to be disassembled are stopped, lifted and fitted in a distinct station.

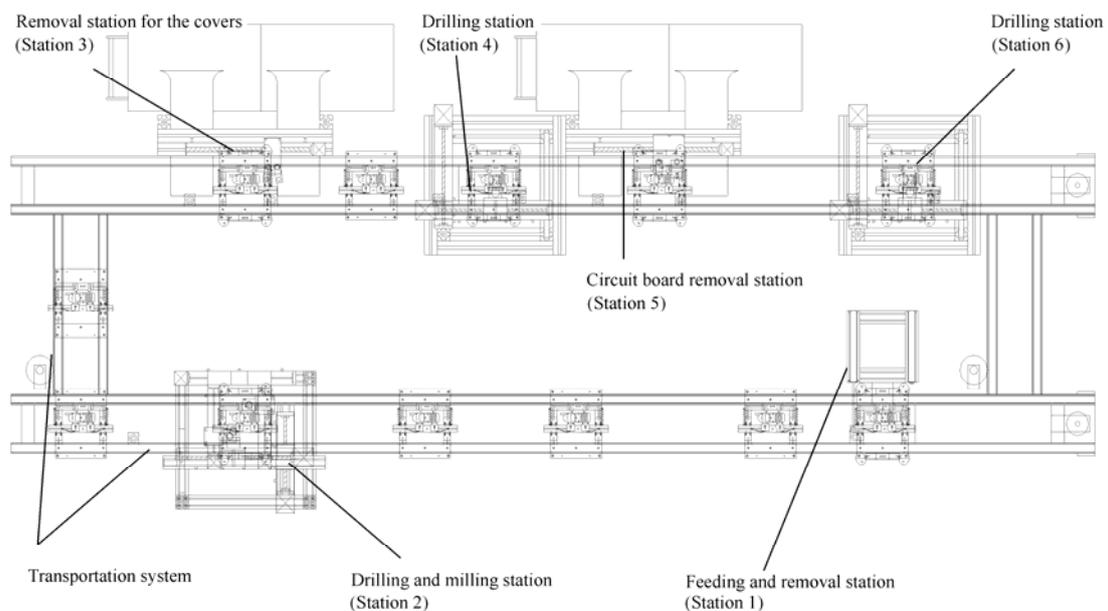


Fig. 2 Layout of the disassembly cell (Kopacek, Kopacek, 2003)

Before the mobile phone is fixed on a pallet the power supply will be removed and the type of the handy will be recognized by a barcode reader manually. Now the control computer knows exactly the type of the handy. The main dimensions of the handy are stored in a database of the host computer.

In the drilling and milling station (no. 2) the upper part of the handy will be cut off from the lower part and the screws – usually between 4 and 17 – are removed by a simple drilling mechanism. The dust content is removed by air from the pallet.

In the third station – the cover removal station – the cover as well as the keyboard of the handy will be removed by pneumatic sucks. These two parts are separated in a storage device. In the next station – drilling station; no. 4 – the screws which connect the printed circuit board on the lower part of the housing are removed. In the printed circuit removal station various other parts will be removed from the handy and separated in special storage devices. Because some mobile phones have additional parts connected with the power part of the housing of the handy the remaining screws will be removed in the last drilling station – station 6. Finally the lower part of the handy will be removed in the fixing and removal station.

As a development of this semi-automated disassembly cell for used mobile phones some previous tests were necessary. For the milling in the drilling and milling station (no. 2) it was necessary to make tests with grinding wheels, with different saws and with milling devices. Finally a milling device was chosen as the right tool for this task.

Further extensive tests were carried out for the removal of the screws. From the literature there are very high sophisticated, complicated and therefore very expensive and heavy devices known. We found a very simple and very cheap method for the removal of the screws.

2.3.A 'Tool Kit' for mobile robots

The basis of a modular concept for mobile robots is the Mobile Robot Platform (MRP) which can be described as a multi-use mobile robot, developed in its basic configuration.

These platforms can be divided in some basic systems:

- Locomotion system
- Driving system
- Main control system
- Communication system

The mobile robot platform can be upgraded and modified by adding a number of peripheral systems and tools for the performance of different tasks or functions. There is a large variety of tools, which can be used.

Basically these tools can be divided into two major categories:

- Conventional tools
- Special tools

Conventional tools (screw drivers, drilling tools, polishing tools, etc) are similar in regard to their function to conventional hand-held tools for manual operations. The difference is in their design, since they have to be fixed on the mobile robot platform, and actuation.

Special tools installed onboard of a mobile robot platform changes the same to a specialized mobile robot system. When special tools are lightweight constructed the manipulation system can be more flexible and with wider reach. Heavier tools cannot be very flexible. They need more rigid and strong manipulation systems. So there is often only one degree of freedom applied, and the other DOFs are realized by the mobility of the platform.

Installing a tool changing system enables the robot to achieve a wide variety of performable operations. Tool changing systems are normally placed at the end of a robot arm. They have to be light, simple and very reliable.

The basic configuration of each mobile robot platform has its integrated sensors. The navigation system makes excessive use of sensor for position determination and collision avoidance. But there are numerous possibilities to upgrade the system with additional sensors for some special applications or to extend its abilities.

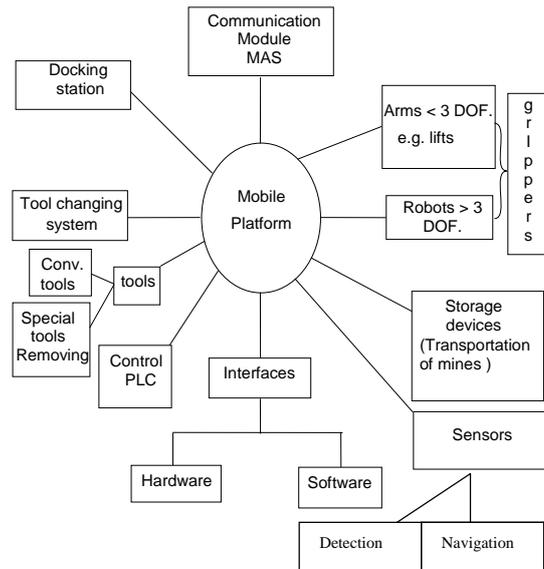


Fig. 3 Modular Robot System (Shivarov, 2001)

In many mobile robot applications transportation is an important part of the overall task. To transport different items mobile robot platforms have to be upgraded with another type of peripheral devices: special storage systems or devices.

Although mobile robot platforms are normally equipped with a communication system it could be necessary to use some special communication systems. Especially in multi agent systems (MAS) where a team of robots acts together is communication between the team members of importance.

2.4. A Tool Kit for Humanitarian Demining (Kopacek, 2004)

According to current estimates, more than 100.000.000 anti-personnel and other landmines have been laid in different parts of the world. A similar number exists in stockpiles and it is estimated that about two million new ones are being laid each year. According to recent estimates, mines and other unexploded ordnance are killing between 500 and 800 people, and maiming 2.000 others per month.

Landmines are usually very simple devices which are readily manufactured anywhere. There are two basic types of mines:

- anti-vehicle or anti-tank (AT) mines and
- anti-personnel (AP) mines.

AT mines are comparatively large (0.8 – 4 kg explosive), usually laid in unsealed roads or potholes, and detonate which a vehicle drives over one. They are typically activated by force (>100 kg), magnetic influence or remote control.

AP mines are much smaller (80-250g explosive, 7-15cm diameter) and are usually activated by force (3-20kg) or tripwires. There are over 700 known types with many different designs and actuation mechanisms.

Hand-prodding is today the most reliable method of mine clearing, but it is very slow, and extremely dangerous. A person performing this type of clearing can normally perform only

this task for twenty minutes before requiring a rest. This method clears one square meter of land in approximately 4 minutes. A better solution for the future is the usage of demining robots.

Today used methods for destroying and removal are brutal force mechanical methods including ploughs, rakes, heavy rolls, flails mounted on tanks. The main problem with this method is the contamination of the ground for 10 – 20 years.

The advantages of robots for demining are:

- Minefields are dangerous to humans; a robotic solution allows human operators to be physically removed from the hazardous area.
- Robots can be designed not to detonate mines.
- The use of multiple, inexpensive robotized search elements minimises damage due to unexpected exploding mines, and allows the rest of the mission to be carried on by the remaining elements.

But there are also disadvantages of using robots:

- it is very difficult for robots to operate in different frequently rough terrain
- they are still expensive
- you need special qualified operators

Teams of robots can be connected so that one team is for searching and one for destroying or displacement. This means that many robots are searching and a few or only one robot is destroying or displacing the mines.

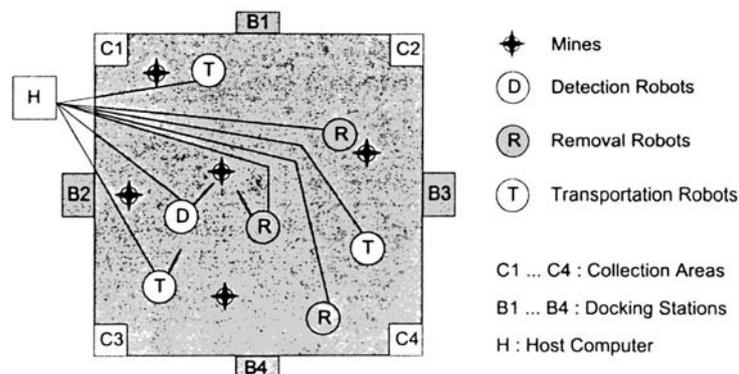


Fig. 4 Robot swarms for demining (Kopacek, 2004)

Robot swarms improve the capacity of robotic applications in different areas where robots are already used today. Robot swarms are similar to – or a synonym for - ‘Multi Agent Systems – MAS’. These systems are very well known in software engineering – “software agents” - since more than twenty years. In the last years there are more and more works related to “hardware agents” like robots.

As mentioned before the use of modular robots is perfect for the design of task specific demining robots because of the similarities between the tasks. All three different types of robots can be constructed by the toolkit mentioned before.

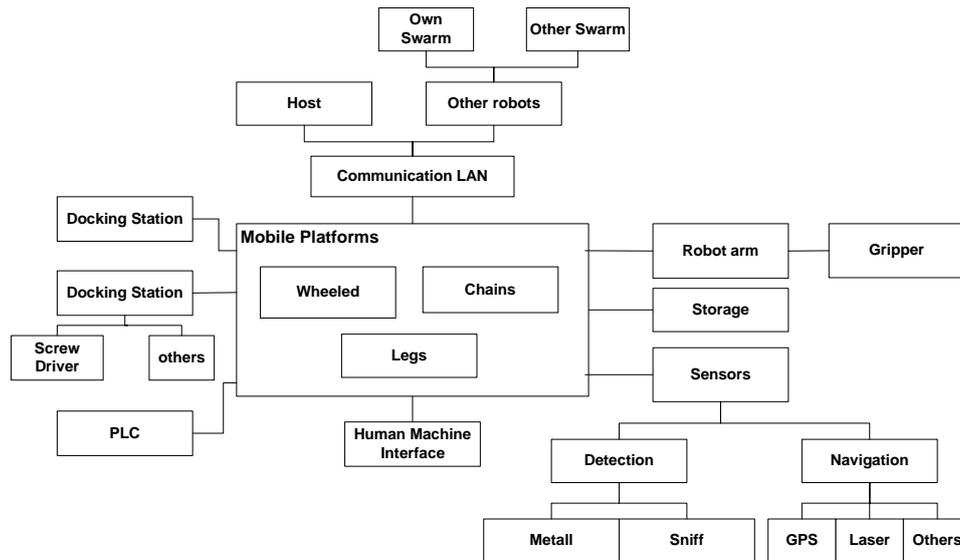


Fig. 5 “Tool Kit” for demining robots (Kopacek, 2004)

2.5.Roby-Run : a mobile mini robot

One of the newest application areas of service robots is the field of entertainment, leisure and hobby because people have more and more free time.

Robot soccer was introduced with the purpose to develop intelligent cooperative multi-robot (agents) systems (MAS). From the scientific point of view a soccer robot is an intelligent, autonomous agent, carrying out tasks together with other agents in a cooperative, coordinated and communicative way. Robot soccer provides a good opportunity to implement and test MAS algorithms. One of the newest approaches in robotics is the application of robots in entertainment, leisure, hobby and education. A new term “edutainment” – composed of two words, education and entertainment, is widely spread. The Robot soccer is a good tool to teach people the complicated technical knowledge in the way of playing. At our institute two robot soccer teams in category MiroSot (Micro-Robot Soccer tournament) are used as a test bed for MAS and edutainment.

The size of playground (Fig. 6) bounded on all sides in category "MiroSot" is 150 x 130cm, 220 x 180cm, 280 x 220 cm or 440 x 280 cm depending on the number of the players.

A camera at a height of approximately 2m delivers pictures to the host computer. With color information placed on the top of the robot, the vision software calculates the position and the orientation of robots and the ball. The host computer generates motion commands using implemented game strategy and position information and sends to the robots by wireless communication.

The human operators are not allowed to directly control the motion of their robots either with a joystick or by keyboard commands under any circumstances. While a game is in progress the host computer can send any information autonomously. The duration of a game shall be two equal periods of 5 minutes each, with a half time interval for 10 minutes.

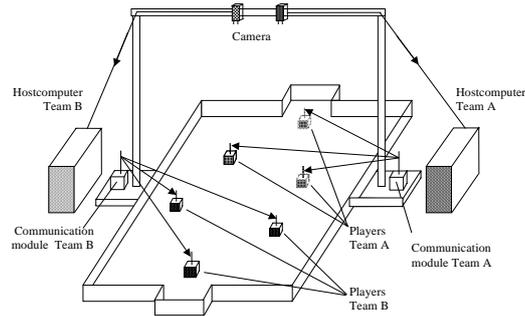


Fig. 6 Overall system of robot soccer (Kopacek et. al, 2003)

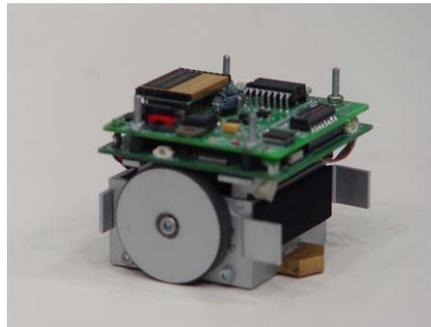


Fig. 7 The mobile mini robot

3. The new concept of a Humanoid Robot

Currently there are worldwide two categories of two legged humanoid robots available:

“Professional” humanoid robots developed by large companies with a huge amount of research capacities. Examples are: the Honda robots (P1, P2, P3, ASIMO) – with the idea to assist humans in everyday working, the SONY robots (SDRX – 3,4,5) – with the background to serve mostly for entertainment, leisure and hobby or in the future as personal robots.

“Research” humanoid robots: There a lot of such robots currently available or in the development stage e.g. approximately worldwide more than 500 University institutes and research centres are active in this field. The robots of this category a usually prototypes developed by computer scientists to implement methods of AI, image processing, theoretical scientists from mechanics implementing and testing walking mechanisms, control scientists to implement new control strategies, social scientists to implement human machine interfaces (HMI) for an efficient communication between humans and humanoid robots.

We are currently working on a humanoid, two legged robot called **ARCHIE**. The goal is to build up a humanoid robot, which can simulate in some situations a human being. Therefore Archie needs a head, a torso, two arms, two hands and two legs and will have the following features:

1. Height: 80 - 100 cm
2. Weight: less than 40kg
3. Operation time: minimum 2hrs

4. Walking speed: minimum 1m/s
5. Degrees of freedom minimum 24
6. “On board” intelligence
7. Hands with three fingers (one fixed, two with three DOFs)
8. Capable to cooperate with other robots to form a humanoid Multi Agent System (MAS) or a “Robot Swarm”.
9. Reasonable low Selling Price – using commercially available standard components.

The control system is realised by a network of processing nodes (distributed system), each consisting of relative simple and cheap microcontrollers with the necessary interface elements. According to the currently available technologies the main CPU is for example a PDA module, one processor for image processing and audio control and one microcontroller for each structural component, e.g.: a Basic Stamp from Parallax.

4. A mobile mini robot for space applications: Roby Space

To get energy from the sun an approach is to set up nets with solar cells in the space and transmit the obtained energy to earth by microwave. A net (approximately 40 x 40 m) equipped with solar cells should be built in outer space (~ 200 km above the earth). The main problem is the positioning of the solar cells on the net structure. To fulfil this task autonomous mobile robots are necessary which are able to move (crawl) on this large quadratic structure – mesh (20 – 40m side length). The distance of the mesh lines is between 3 and 5cm; the thickness of the wires between 1 and 3mm.

The features of an autonomous mini robot for this purpose are:

- the maximum dimension 10x10x5cm
- light weight (less than 1 kg)
- on board power supply for approximately 10min
- equipped with a camera sending pictures to the earth
- wireless communication with the mother satellite by Bluetooth or similar
- free movement on the mesh
- mechanical and electronic robustness against vibration and shock.

The paper deals with the development of a low cost mini robot able to move on a mesh in outer space. Main difficulty is the design of the moving and holding mechanism of the robot on the mesh. Other difficulties are that the working environment is outer space (about 200 km over the earth) – low/high temperature, radiation, micro-gravity etc. - and the robot should be brought by means of rocket - vibration and shock. The requirements on the robot are the limited maximum size (10 x 10 x 5 cm), a simple mechanical construction, miniaturized electronics, robustness, “low cost”, and independence of the mesh’s dimension (from 3 x 3cm to 5 x 5cm).

Already two prototypes (Fig. 8) - Roby-Sandwich and Roby-Insect - were built and tested. Two tests – low temperature test at the 40 degree below zero and micro gravity tests- were already successfully done. At the 40 degree below zero Roby-Sandwich crawled on the net without any problem. In January 2005 two robots were tested in the micro-gravity environment by means of the parabolic flights in Japan.

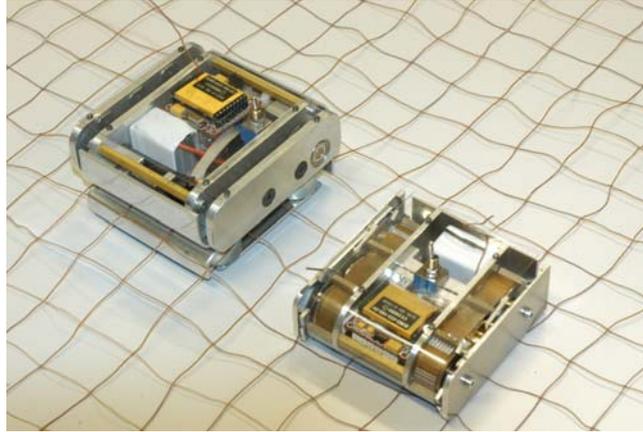


Fig. 8 Roby-Sandwich (Left) and Roby-Insect (Right) (Kopacek et. al (2004))

5. SUMMARY AND OUTLOOK

In this paper some new applications of a new robot generation are described. In addition modern information technologies lead to loneliness of the humans (teleworking, telebanking, teleshopping,...). Therefore service robots will become a real “partner” of humans in the nearest future. One dream of the scientists is the “personal” robot. In 5, 10 or 15 years everybody should have at least one of such a robot. Because the term personal robot is derived from personal computer the prices should be equal.

6. LITERATURE

Kopacek,P. and D.Noë (2001): Intelligent, flexible assembly and disassembly. In: *Proceedings of the IFAC Workshop „Intelligent Assembly and Disassembly – IAD’01”*, p.1–10.

Kopacek,P. and B. Kopacek (2003): Robotized disassembly of mobile phones. In: *Preprints of the IFAC Workshop „Intelligent assembly and disassembly – IAD’03”*, Bucharest, p.142-144.

Kopacek, P., M.-W. Han, B. Putz, E. Schierer, M. Würzl (2003). Robot soccer a first step to “edutainment” In *Proceedings of RAAD’03, 12th International Workshop on Robotics in Alpe-Adria-Danube Region Cassino, Italy*

Kopacek, P. (2004). Robots for humanitarian demining. In: *Advances in Automatic Control (Ed. M Voicu)*, Kluwer, p.159- 172.

Kopacek, P. M. Han, B. Putz, E. Schierer, M. Würzl (2004): "A concept of a high-tech mobile minirobot for outer space"; in: "ESA SP-567 Abstracts of the International Conference on Solar Power from Space SPS'04 together with the International Conference on Wireless Power Transmission WPT5", Granada, Spain, 132 - 133.

Shivarov, N. (2001): A ‘Tool Kit’ for modular, intelligent, mobile robots, *PhD. Thesis*, Vienna University of Technology.

Acknowledgement: Parts of this project is supported by ESA under ESTEC/Contract No.18178/04/NL/MV- Furoshiki Net Mobility Concept

High Speed Robot Vision Authentication

Wael Adi¹, Ali Dawood^{*}, Ali Mabrouk[°]

^{*},¹Etisalat College of Engineering, United Arab Emirates.

[°]Lufthansa Systems AS, Germany.

email: wael@ece.ac.ae, dawood@ece.ac.ae, ali.mabrouk@lhsystems.com

Abstract: *Image information streams involve relatively high-speed large data packets for most applications. To secure ad-hoc transactions like those involved in Robot environment, public-key techniques with high-complexity modular computations are required. This paper presents a general technique to simplify processing modular arithmetic for image authentication in environments having low computation capability such as robot equipments. We introduced earlier a general novel technique called Fuzzy-modular computation in [1]. The basic idea behind such fuzzy arithmetic is a rough execution of modular computations at sites lacking computational power. The result is a non-completed job, which need to be completed at the end-user site to obtain the exact results. It was shown that a minor security loss would result out of this fuzzy computation. In this work, a similar technique however for fast public-key image authentication is proposed using fuzzy-computations applied to El-Gamal authentication mechanism. The fuzzy operations are embedded carefully in the discrete exponentiation computations to authenticate real time images without performing any real time exponentiations. This results with highly simplified processing at one site (like mobile equipment site) allowing relatively high-speed processing at low-power environment. The technique is rather general and can be employed similarly for other security mechanisms requiring shifting computation complexity from one particular weak system units to another powerful unit.*

1. INTRODUCTION

Robots are mobile equipment with mostly limited computation power. The need for secured transactions with Robots is rapidly growing with the increasing robot intelligence and capabilities. A common problem in authentication over open channel is to sign the source images coming from a mobile device or a robot. Possible application is to assure that image received from certain robot device is a true fresh capture from the device itself and not a playback of a recorded image or is coming from another intruding robot.

One possible application for such source-authenticated images is to check the identity of a person observed by a robot looking securely online at his image coming from a remote site. This would allow authenticated real-time checking of a visitor's identity and even signature related to his authentic image. If the image tends to be authentic, then this operation would be somehow equivalent to a *personal* checking of user's identity when the user presents his ID card. The user should be in that case a certified system participant. Variety of other application scenarios can be derived from certified real time images.

¹ Author is on leave from the technical University of Braunschweig, Germany

Image data need mostly to be processed under very limited timing conditions. Therefore image security processing requires often very high computational power. The aim of this work is to reduce the processing power required for image traffic by using our fuzzy modular computation formerly proposed in [1]. Image data are very large. If a digital data frame is to be signed using El-Gamal signature scheme, then the main online operation required is to compress the data and apply some multiplication with precompiled data. The proposed technique of this work would use a simplified fuzzy computation model for digitally signing one image frame. Fuzzy computation require however more signature space. As image data are inherently large such additional space required is not a serious disadvantage of our proposed scheme. Authenticating a JPEG compressed image frame is taken as an example format in our presentation. Contemporary mobile systems do not use public key systems still due to the lack of computation power [2]. Many modern authentication standards are still using secret key schemes as that used for 3GPP mobile system [3].

The rest of this paper is organized as follows: Section 2 describes the robot vision scenario and some corresponding technology requirements and applications. Section 3 describes the fuzzy processing model and its complexity, Section 4 details the security threats and possible attacks and Section 5 presents a summary and conclusion.

2. AUTHENTICATED ROBOT VISION

Robots are autonomous machines with high degree of freedom. Autonomous robot vision frames could represent very sensitive information source, which require authentication to be usable for many applications. Fig. 1 shows a possible simple application scenario where the robot should send observed image frames through a wireless channel.

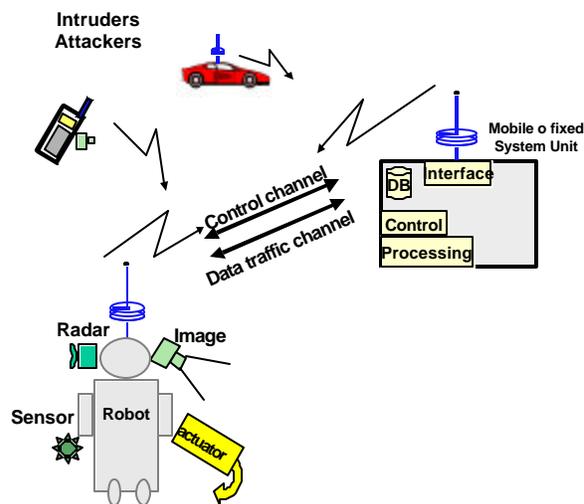


Figure 1: A scenario for authenticated robot image frames

The image frames are sent over a wireless channel. To avoid substituting such images by other intruding attacker, the image frames should be authenticated such that no single image can be accepted at the receiving end without being trustable. The application spectrums for such authenticated transfer are very wide. The operation should be processed in real time. The objective of this work concentrates on making such highly complex authentication possible in real time with limited available computational power of most robots in use. For example the robot could be a guard observing a building and moving autonomously around.

2.1 Unique Identity Requirement

The robot is seen as an individual with its unique identity. Human beings have got uniqueness through DNA signature. Robots need something similar. We propose the use of our write only memory technique described in [5], [6] for generating “physically-unique” identity for a robot. Fig. 2-1 shows the principle operation required to personalize robots. Once a robot is personalized, it is possible to remotely and securely check its identity. The robot manufacturer could be the one who digitally signs the identity and be responsible for uniqueness. Other scenario for building uniqueness is certainly possible depending on the background of the expected application and security level.

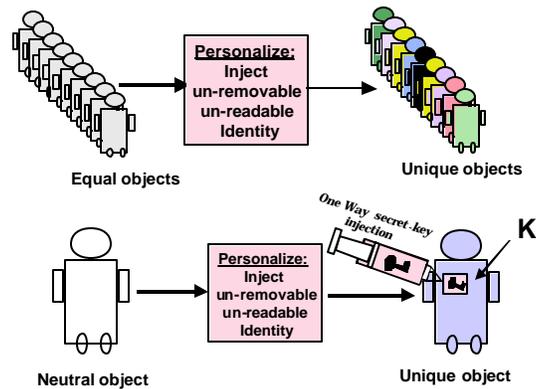


Figure 2-1: Robot unique identity and personalization

The unique secret device identity key SDI is stored in a core unit in the robot, which should be physically only removable if the robot is functionally destructed. This identity is used completely or partially as a secret key for generating the required signature.

Fig. 2-2 shows a possible implementation scenario by embedding/integrating a non-removable Identity Module (IM) in a core hardware element in the robot main processing unit as its main memory. A unique secret device identity SDI is stored in a write only memory element. SDI, is usually generated by using a keyed hash function where $SDI = F(DI, MK)$. DI is the clear device serial number and MK is the secret master key of the robot personalizing authority. The function F is a highly secured hash function such as a good block cipher. The input CH is to be used, as a challenge to the robot by some random number and RES is the corresponding response to the

verifier. The identity module infrastructure allows authenticating the robot physically to avoid robot duplication. Once the robot is proved to be unique the data transmitted as image from the robot can be jointly authenticated using the assigned public-key signature strategy.

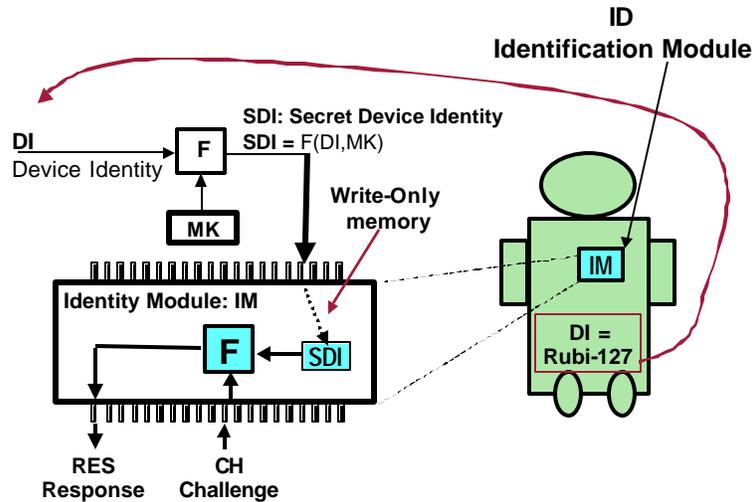


Figure 2-2: Robot personalization technology

3. IMAGE SIGNATURE STRATEGY

To avoid any kind of image tampering the signature need to be derived directly from the image data itself. Moreover, since JPEG compression algorithm is a lossy image compression technique, therefore the compressed data should be used instead of the original pixel values so that signature can be reproduced and verified at the receiving site.

In JPEG algorithm the image is divided into blocks of 8x8 pixels and each block undergoes a DCT process, which produces 64 DCT coefficients [4]. The quantized version of such coefficients will be used to compute the signature as shown in the processing graph of figure 3.

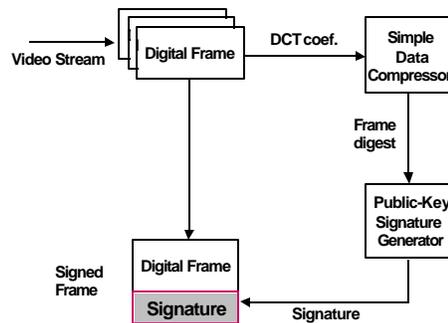


Figure 3: Possible Image Authentication Strategy

3.2. Line based Signature

In this approach, the last block of every row will be used to carry 64-bit signature of that particular row. For example, with reference to the $N \times M$ frame in fig. 4, information of $D_{1,1}, D_{1,2}, \dots, D_{1,M-1}$ will be used to compute first row's signature S_1 , and it replaces $D_{1,M}$ block. This is illustrated in figure 6. Remaining rows will be signed in a similar way. Fig. 7 shows a signed image following such approach. The advantage of such approach is that every row of blocks can be authenticated independently from the other rows and any false image can be detected earlier after checking few rows. In other words, there is no need to wait for the whole frame's data to authenticate the image frame.

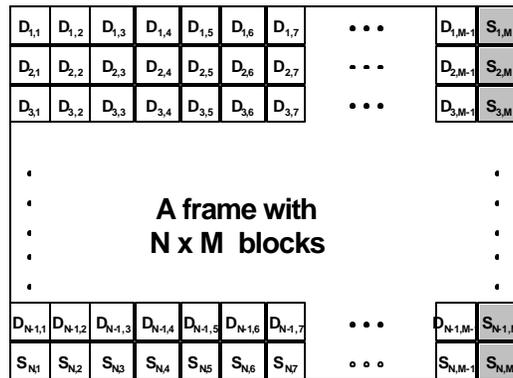


Figure 6: Line based signature allocation.

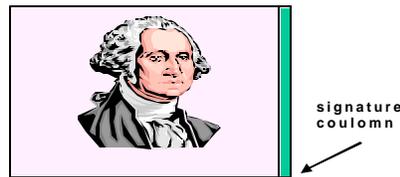


Figure 7: Line based image signatures

A robust authentication can be achieved when a combination of both approaches is considered.

4. OPEN-KEY SIGNATURE COMPUTATIONS

The algorithm is a possible variant of the El-Gamal scheme as a public-key digital signature scheme. The signature mechanism requires first compressing the image message using a hash function H . Figure 8 shows the basic set-up of the El-Gamal-based digital signature algorithm.

The robot as user A having a private key $X_a=f(K_i)$ signs a binary message M of arbitrary length. Any entity B can verify this signature by using A's public key Y_a . The signing and verification procedures are described below:

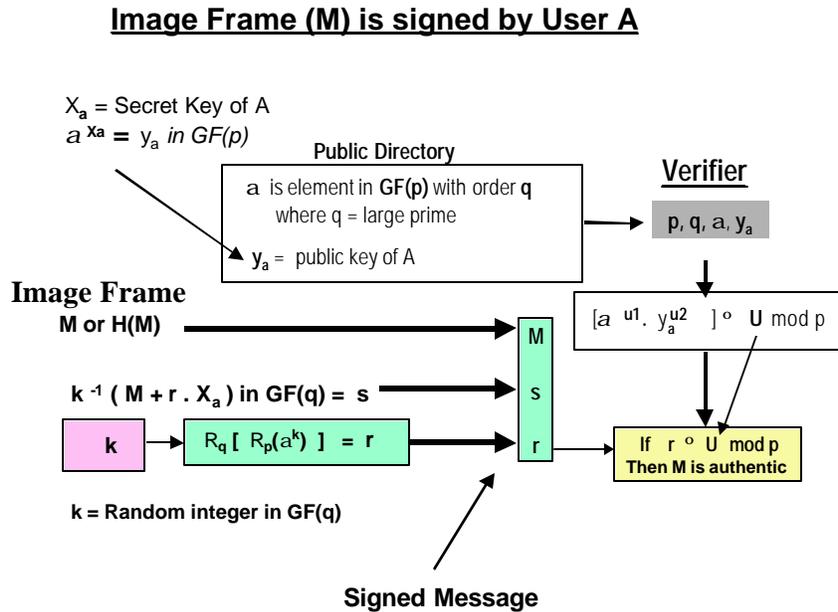


Figure 8: El-Gamal based Digital Signature Scheme

A) Signature generation:

The User A should do the following:

- i. Select a random secret integer k such that $0 < k < q$.
- ii. Compute $r = (\alpha^k \text{ mod } p) \text{ mod } q$.
- iii. Compute $k^{-1} \text{ mod } q$
- iv. Compute $s = k^{-1} (M + r X_a) \text{ mod } q$.
- v. A's signature for M is the pair (r, s) .

B) Verification:

To verify A's signature (r, s) for the message M, user B should do the following:

- i. Obtain A's authentic public key Y_a .
- ii. Verify that $0 < r < q$ and $0 < s < q$; if not, then reject the signature.
- iii. Compute $w = s^{-1} \text{ mod } q$ and $H(M)$ and let the size of $H(M)$ in bits be equal to the bit-size of q . The standard functions as those of MILENAGE proposed and deployed in mobile systems can be used [4].
- iv. Compute $u1 = w.H(M) \text{ mod } q$ and $u2 = r.w \text{ mod } q$

- v. Compute $U = (\alpha^{u_1} \cdot y^{u_2} \text{ mod } p) \text{ mod } q$.
- vi. Accept the signature if and only if $r = U \text{ mod } p$.

As it could be seen from the above set-up, signing a piece of data M involves various modular operations. Some of these operations are of high bit complexity (Table 1).

	Signature generation	Bit complexity
Modular Addition	1	$O(\lg n)$
Modular Subtraction	0	$O(\lg n)$
Modular Multiplication	2	$O((\lg n)^2)$
Modular Inversion	1	$O((\lg n)^2)$
Modular Exponentiation	1	$O((\lg n)^3)$

Table 1: Bit Complexity of modular operation

5. FUZZY IMAGE SIGNATURE MECHANISM

The processing power of the devices used in wireless communication is considerably limited; this represents a bottleneck for real-time and online operations such as signing an image for mobile transactions. The Fuzzy Modular Mechanism can be used to reduce the complexity of the image securing functions.

Considering a public-key signature for a full image frame in a mobile device we propose the following protocol scenario using low-complexity fuzzy computations.

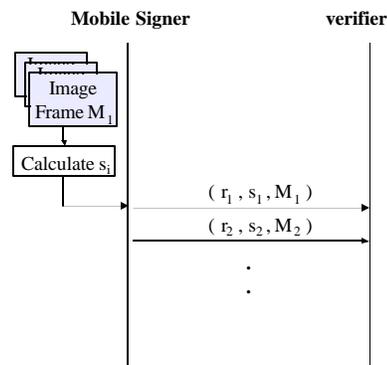


Figure 9: proposed authentication protocol

Figure 9 shows the main transactions required to run the signature protocol. It is a one-pass protocol. For security reasons, a secret integer k has to be chosen at random and should be used only once. For the sake of acceleration, k and k^{-1} should be pre-computed **offline**. The user could hold a list of randomly selected integers together with their modular inverse.

The main parts of the signature generation are the following:

A. Off-line Computing of $r \equiv (a^k \bmod p) \bmod q$

This operation is performed off-line when the mobile device is idle. A list of several k 's and the corresponding r 's should be kept as (k,r) pairs in the device memory for later use. This operation may take a long time however it has no influence on the signature speed as it is running off-line. The price for that is to occupy the mobile memory with several (k,r) pairs. This is not considered as a great disadvantage as memory space is usually not a bottleneck in the mobile device.

B. Online Computing of $s = k^{-1} (M + r X_a) \bmod q$:

The goal is to reduce the complexity of the following computation:

$$s \equiv k^{-1} (M + r X_a) \bmod q \quad (1)$$

Equation (1) can be rewritten as follows:

$$s \equiv [k^{-1} M + k^{-1} r X_a] \bmod q \quad (2)$$

A good approach here is to use pre-computations. The user has just to do the following steps once offline:

- Compute $k^{-1} \bmod q$ and save it for later use.
- Compute $(k^{-1} X_a) \bmod q$ and save it for later use.

Such pre-computations can preferably be attached to the first off-line set-up operation (A) shown above.

For signing a piece of image data M or precisely $H(M)$, the user has to execute the following:

- Compute $A = k^{-1} \cdot H(M) \quad (3)$

- Compute $B = r \cdot (k^{-1} X_a) \quad (4)$

- Compute $s = A + B - t \cdot q \quad (5)$

Signature memory complexity:

Notice that no modular computations are involved.

The signature tuple $\mathbf{S} = (r, s)$ is then sent to the verifier. The size of \mathbf{S} is roughly $|\mathbf{S}| = |r| + |s|$ as it is a concatenation of the vectors r and s . Assuming that the modulus q size is $n = \log_2 q$. From (5) the size of s is roughly $\text{MAX}[|A|, |B|]$. From (3) and (4), $|A|$ or $|B|$ could be at most $2n$ bits. The weight of t is assumed to be $\log_2 q / 2$ as required in [1], therefore the bit-size of $t \cdot q$ is upper bounded by $2n$ bits. $|s|$ is then at most around $2 \log_2 q = 2n$ bits.

For practical applications the size of $q = n = \log_2 q \approx 1000$ bits, thus the number of bits required to sign a frame is $3n = 3000$ signature bits. The extra redundant signature bits due to fuzzy computations

are only n bits that is only 1000 bits in that example. This amount of size increment is negligible compared to the bulk data of a single image frame. The signature can be accommodated in just one image row as the last row of a frame as shown in Fig 4.

The storage of a pre-computed B having $2n$ bits is also required *for each transaction*. As for security reasons the random number k and the corresponding $r = (\alpha^k \bmod p) \bmod q$ should not be repeatedly utilized.

Signature computation complexity:

The complexity of the signature generation for a single image frame M would require only the computations shown in equations (3), (4) and (5). This includes only two multiplications and two additions. We are assuming here that computing $H(M)$ by using for example SHA-1 algorithm can be processed online any time loss and B is a pre-computed value. If we assume having a fuzzy modular multiplier as in [1], then only one fuzzy multiplication is required to generate the signature

Assume that we have q as a prime in the range of 1000 bits, only two fuzzy modular multiplications, which are equivalent to 2 usual multiplications as shown in [1].

6. SECURITY THREATS AND POSSIBLE ATTACKS

Breaking the system is equivalent to breaking the fuzzy system described in [1].

The fuzzy computation is only in $s = A + B - t' \cdot q$

As t' is to be selected having the weight $n/2$.

The overall security loss due to fuzzy computations is then very minor and equivalent to that computed in [1]. That is the strength of the system key represented by the size of q in bits $= n = \log_2 q$ is reduced by $\log_2 n$ bits. The new system key-size n' is then

$$n' = \log_2 n - \log_2 (\log_2 q) \quad (6)$$

For example if the key size (bit-size of q) is $n=1000$ bits then the new key-size $n' = 1000 - \log_2 1000 = 990$ bits. This is a negligible security reduction.

7. SUMMARY AND CONCLUSION

The proposed system offers a novel technique having practical impact on the usage of robot systems in public-key security environment. The fuzzy modular computations allow reducing the computation power required for public key security functions by avoiding any online modular multiplication and modular exponentiation computations.

8. REFERENCES

- [1] W. Adi , Fuzzy Modular Arithmetic for Cryptographic Schemes with Applications to Mobile Security EUROCOM 2000
- [2] Technical Specification 3G Security, Security Architecture 3G TS 33.102 V. 3.2.0 from 10.1999
- [3] Specification of the MILENAGE Algorithm Set. 3GPP TS 35.206 V5.0.0 ETSI, <http://www.3gpp.org>, 2002.
- [4] G. K. Wallace, "The JPEG still picture compression standard", Communication of the ACM, Vol. 34, No. 4, pp.30-44, April 1991.
- [5] W. Adi, "Secured Mobile Device Identification with Multi-Verifier", International Conference on Telecommunications (ICT2001, 2001), pp. 289 – 292
- [6] W. Adi, European Patent, " METHOD FOR THE CRYPTOGRAPHICALLY VERIFIABLE IDENTIFICATION OF A PHYSICAL UNIT IN A PUBLIC, WIRELESS TELECOMMUNICATIONS NETWORK, EP000001 290905A2, Priority date 28.03.2001.

CONTINUOUS GAIN SCHEDULING FOR UNDERACTUATED SYSTEMS

Y.Tachwali^{*}, M.A.Jarrah^o

^o Mechatronics Program, American University of Sharjah. Director and Professor

^{*} Mechatronics Program, American University of Sharjah. Student
email: tachwali@ieee.org, mjarrah@ausharjah.edu

Abstract: *In this paper we discuss an implementation of continuous gain scheduling to stabilize an underactuated system chosen to be the Pendubot setup. The goal of our controller is to bring the Pendubot to its middle equilibrium point and then stabilize the free rod in the vertical position while moving the actuated rod from the bottom position to the upper position and bringing it down again completing full rotation around the actuated point. A continuous gain scheduling controller is presented to produce acrobatic maneuvers by the Pendubot. Simulation results show the functionality of this control method and its effectiveness to overcome the singularity gain conditions at the horizontal position.*

1. Introduction:

Underactuated systems are defined as systems that have fewer control inputs than degrees of freedom and arise in applications, such as space and undersea robots, mobile robots, walking, brachiating, and gymnastic robots. The class of underactuated mechanical systems is thus rich in both applications and control problems [1].

One of these systems is the Pendubot, which is a two-link planar robot with actuator at the shoulder (link 1) and no actuator at the elbow (link 2), which moves freely around link 1. This system is a simple underactuated mechanical system that shows second-order nonholonomic properties.

Many contributions have been presented to stabilize the system in its upright position or in the middle position, [2] developed an intelligent controller on the basis of Genetic Algorithms. First they develop a static State Feedback Controller with inner non-collocated partial feedback linearization loop to stabilize the nominal system.

In [3] the balancing control is based on an energy approach and the passivity properties of the system. The control strategy is based on an energy approach and the passivity properties of the Pendubot. A Lyapunov function is obtained using the total energy of the system. The analysis is carried out using the LaSalle's theorem.

A hybrid controller for feedback stabilization of the Pendubot was presented [4]. The hybrid controller was constructed based on a general form originally created for feedback stabilization of a class of underactuated mechanical systems.

Other papers showed a control method to move the Pendubot in a certain path.[5] has presented acrobatic control. They have designed a fuzzy controller for keeping first link swinging periodically while the second link maintains standing vertically, and derived an LMI design condition to realize acrobatic behavior. After selecting reference fuzzy model dynamics, an acrobatic controller has been designed by solving the LMI condition.

Similar work has been presented in [6] where a T-S fuzzy scheme for trajectory tracking of underactuated mechanism has been implemented to track sinusoidal signals of big amplitude. Research is being pursuing in order to solve the problem of Pendubot trajectories crossing singular points.

The present work proposes control logic based on continuous gain scheduling to stabilize the Pendubot through the singular points. Figure 1 shows system overview.

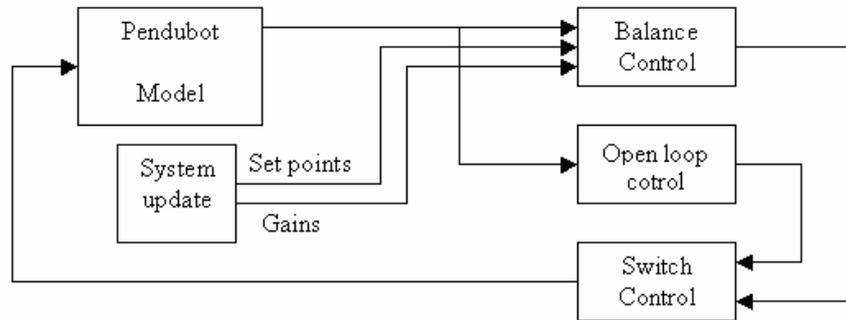


Figure 1. System Overview

2. System Dynamics [7]:

Consider the two links underactuated planar robot, called the Pendubot. We will consider the standard assumption, i.e. No friction, etc...

- m_1 : mass of link 1
- m_2 : mass of link 2
- l_1 : Length of link 1
- l_2 : Length of link 2
- l_{c1} : Distance to the center of mass of link 1
- l_{c2} : Distance to the center of mass of link 2
- I_1 : Moment of inertia of link 1 about its centroid
- I_2 : Moment of inertia of link 2 about its centroid
- G : Acceleration due to gravity
- q_1 : Angle that link 1 makes with the horizontal
- q_2 : Angle that link 2 makes with link 1
- τ_1 : Torque applied on link 1

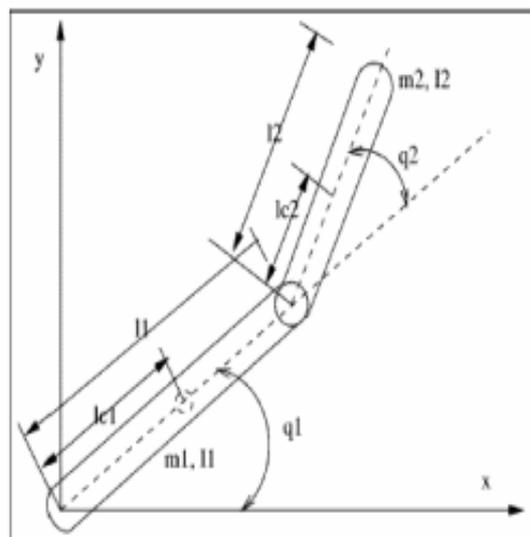


Figure 2. The Pendubot System

We have introduced the following five parameters equations:

$$\begin{aligned}
\theta_1 &= m_1 l_{c1}^2 + m_2 l_1^2 + I_1 \\
\theta_2 &= m_2 l_{c2}^2 + I_2 \\
\theta_3 &= m_2 l_1 l_{c2} \\
\theta_4 &= m_1 l_{c1} + m_2 l_1 \\
\theta_5 &= m_2 l_{c2}
\end{aligned} \tag{1}$$

For control design we neglect the friction, these five parameters are all that are needed.

2.1. Equations of motion via Euler – Lagrange formulation:

Lagrange's Equation is given by the following :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}}(q, \dot{q}) \right) - \frac{\partial L}{\partial q}(q, \dot{q}) = \tau \tag{2}$$

where $q = (q_1, \dots, q_n)^T$ represents the generalized variables, one for each degree of freedom of the system, $\tau = (\tau_1, \dots, \tau_n)^T$ denotes forces that are externally applied to the system. In our case, the generalized variables are q_1 and q_2 , i.e. $q = (q_1, q_2)^T$ and $\tau = (\tau_1, 0)$, where τ_1 is the force applied on the first link.

Lagrangian function is equal to difference between the kinetic energy and the potential energy:

$$\begin{aligned}
L = K - P &= \frac{1}{2} (\theta_1 + \theta_2 + 2\theta_3 \cos q_2) \dot{q}_1^2 + \\
&\frac{1}{2} \theta_2 \dot{q}_2^2 + (\theta_2 + \theta_3 \cos q_2) \dot{q}_1 \dot{q}_2 - \theta_4 g \sin q_1 + \\
&\theta_5 g \sin(q_1 + q_2)
\end{aligned} \tag{3}$$

2.2. Lagrangian Matrices:

The following is the Lagrangian formulation of the dynamics of an n-degree of freedom mechanical system:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = B(q)\tau \tag{4}$$

Where $q \in R^n$ is the vector of generalized coordinates, $\tau \in R^m$ is the input generalized force ($m < n$), and $B(q) \in R^{n \times m}$ has full rank for all q .

The previous formula can be partitioned [8] into two sets of equations by partitioning the vector $q \in R^n$ of the generalized coordinates as $q_1 \in R^l$ and $q_2 \in R^m$, where $q_1 \in R^l$ represents the unactuated (passive) joints and $q_2 \in R^m$ represents the actuated (active) joints. In our Pendubot application the system has $n=2$ joints and $m=1$ activated joint.

The following is a brief description of each part of Lagrangian formulation:

$$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix}$$

where q represents the angle vector, τ represents the input vector, which is the torque applied on the actuated joint while the second joint is free.

$$D(q) = \begin{bmatrix} \theta_1 + \theta_2 + 2\theta_3 \cos q_2 & \theta_2 + \theta_3 \cos q_2 \\ \theta_2 + \theta_3 \cos q_2 & \theta_2 \end{bmatrix} \quad (5)$$

Is the mass matrix.

$$C(q, \dot{q}) = \begin{bmatrix} -\theta_3 \sin(q_2) \dot{q}_2 & -\theta_3 \sin(q_2) \dot{q}_2 - \theta_3 \sin(q_2) \dot{q}_1 \\ \theta_3 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \quad (6)$$

Includes the Coriolis, friction, and centrifugal terms.

$$g(q) = \begin{bmatrix} \theta_4 g \cos q_1 + \theta_5 g \cos(q_1 + q_2) \\ \theta_5 g \cos(q_1 + q_2) \end{bmatrix} \quad (7)$$

Contains the terms derived from the potential energy, such as gravitational and elastic generalized forces.

Note that the mass matrix $D(q)$ is symmetric and positive definite for all q . Moreover, the potential energy of the Pendubot can be defined as

$$P(q) = \theta_4 g \sin q_1 + \theta_5 g \sin(q_1 + q_2) \quad (8)$$

Note that P is related to $g(q)$ as follows:

$$g(q) = \frac{\partial P}{\partial q} = \begin{bmatrix} \theta_4 g \cos q_1 + \theta_5 g \cos(q_1 + q_2) \\ \theta_5 g \cos(q_1 + q_2) \end{bmatrix} \quad (9)$$

2.3. Pendubot Linearization at the equilibrium points:

The purpose of linearizing the model is for the purpose of designing a controller. A linear model can be put into state space representation where a Linear Quadratic Regulator (LQR) controller can be designed in order to stabilize the Pendubot

2.3.1 To find the equilibrium points for $\tau = 0$

$$q_1 = \frac{\pi}{2} + k_1 \pi \quad (10)$$

$$q_2 = k_2 \pi$$

This shows repeated possible solutions for two positions, which are the up right position and the mid position.

2.3.2 To find the equilibrium points for $\tau \neq 0$

Then we will have an infinite number of equilibrium points that satisfy:

$$q_1 + q_2 = \frac{\pi}{2} + k_1\pi$$

$$q_1 = \cos^{-1}\left(\frac{\tau}{\theta_4 g}\right)$$
(11)

When we linearize the model around one of the equilibrium points we have to test the controllability of the system by calculating $\det(B|AB|A^2B|A^3B) \neq 0$ and a full state feedback controller $\tau = -K^T X$ with an appropriate gain vector is able to successfully stabilize the system at the equilibrium.

3. Proposed Controller design:

The controller will be composed of two parts, one for achieving regular tracking control using continuous LQR gain scheduling method for the whole circle path except the neighborhood of the singular point where another controller will apply a small opposing torque to slow down the falling of actuated rod at this region, which was selected to be between -0.12π and $+0.12\pi$. In order to pass this region safely, a dedicated logic was performed to handle the switching to the tracking controller successfully. The LQR gains and set points are updated to the controller online at the rate equal to the sampling time of the Simulink® simulation.

3.1. Tracking controller design:

First we determine a number of equilibrium points from $-\pi$ to $\pi/2$ with a step of $\pi/1440$, these equilibrium points are determined by (14). The torque required for each equilibrium position has been calculated and figure 3 is showing the relation between the torque and the actuated angular position. It is clearly shown that the maximum torque required is 9.6 at the horizontal position while the up and mid position does not need any torque to be generated from the actuated joint to hold the free rod in the steady state.

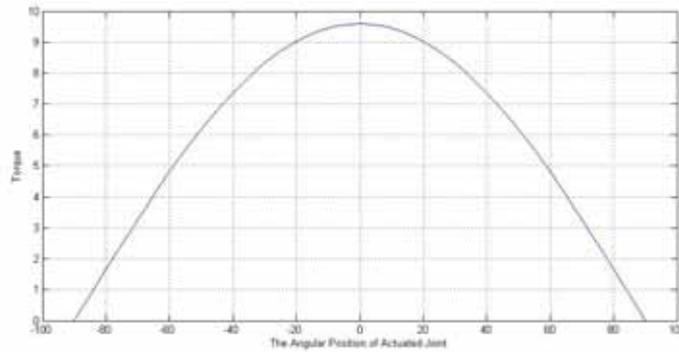


Figure 3. Torque Applied At Equilibrium Points

Then state feedback gains can be found using LQR method, These values were determined using the rule of thumb mentioned in [9] where each value in the diagonal line is equal to $\frac{1}{\Delta X_{\max}^2}$, therefore $Q(1,1)$ and $Q(1,3)$ where determined by setting the limitation of one step in the position $= \pi/1440$ away from the varying set point, a value of 210 will be required.

However, Iterations in simulation showed that the value of 153 is enough for achieving good tracking.

$$Q = \begin{bmatrix} 153 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 153 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

On the other hand, $Q(2,2)$ and $Q(4,4)$ have to set some limitations on the speed in order to break down the pendulum while crossing the singular point, however large values of $Q(2,2)$ and $Q(4,4)$ will increase the encoder noise sensitivity. So we decided to put the maximum speed is the speed at which the actuated rod in passing the boundaries of the open loop control which is approximately 1 rad/sec.

The cost or controller effort matrix R has been set to a fixed value of 0.01. This value was determined after several iterations in simulation, where the actuator has a maximum torque of 10, so in the same manner R was calculated by [9]

$$R = \frac{1}{U_{\max}^2} \quad (13)$$

Figure 4 shows the calculated gains using LQR method.

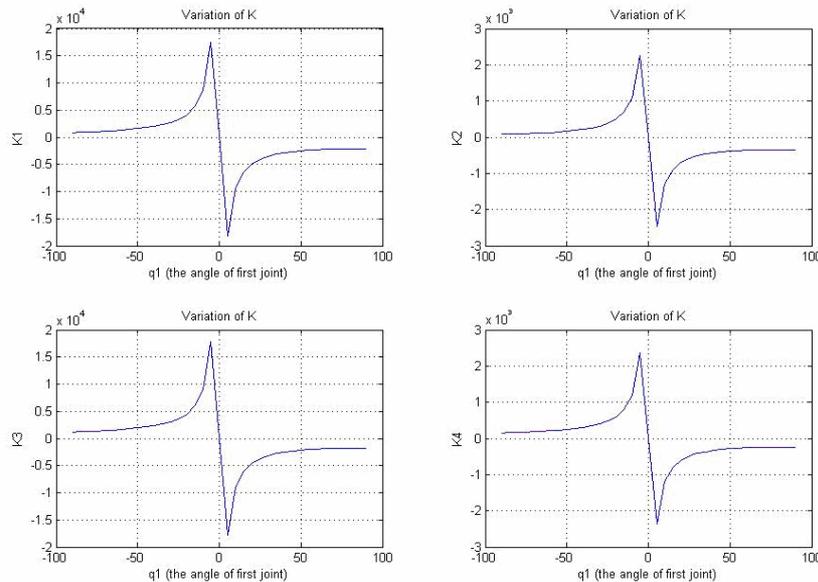


Figure 4. State Feedback Gains

A sharp transfer from positive to negative is taking a place at angle zero which is the singular point at the horizontal position. Passing this point is a critical operation due to the large gain values required at the neighborhood of this point. Moreover, catching the rod back in the opposite quarter is not an easy issue since the attraction basin is shrinking while approaching this singular point and this is natural due to the dramatic increment in gains required to achieve the balance.

3.2. Switch controller:

Linearizing the system around the horizontal position $(0, \pi/2)$ showed that it is uncontrollable due to the singularity in the controllability matrix $[B \ AB \ A^2B \ A^3B]$. Therefore it does not make sense to apply a close loop control at this point.

3.2.1 Determining the open loop control region:

A small part of the previous gains was saturated in the region $[-0.1\pi \text{ to } +0.1\pi]$ since they will not be used in the control (the same is applied to the related set points). Instead, we made the open loop controller to apply a small opposing torque to slow down the falling action at this point. This action has to be in advance before the sudden switch in the gain and set points from 0.1π to -0.1π . Therefore the controller take the action at an envelope of $\pm 0.12\pi$ and at the moment of crossing the boundary of -0.12π the same set point will be applied to it. See figure 8.

3.2.2 Improving the controllability at the singularity region :

The free rod has to tilt a small angle away from the vertical position towards the centre of rotation to overcome the uncontrollability feature at this position, This action will remove the singularity from controllability matrix and can be achieved by making a spike in the gain before passing this region. This spike is created by making a sudden decrement in the gain which will lead to a sudden small fall in the actuated rod enough to create the required tilt in the free rod. See figure 10.

3.2.3 Catching the falling rod and switching to tracking control:

The actuated rod will pass the region and reach a speed of 1 rad/sec. The normal tracking controller performance is excellent for small angular velocities for both joints, going beyond the value of 1 rad/sec will create a strong action in the actuated joint to obey the performance role determined by Q. However this strong action was decreased by a factor of 0.02 if the speed becomes greater than 1 rad/sec. Figure 14 shows the switching action between normal control action and the decreased action. Figure 5 shows the sequence of controllers being applied:

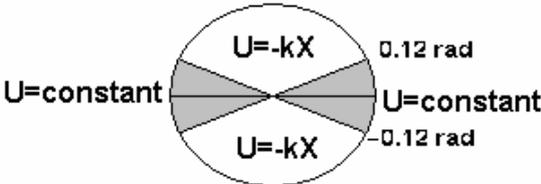


Figure 5 Control Action and Tracking Path

4. Program structure:

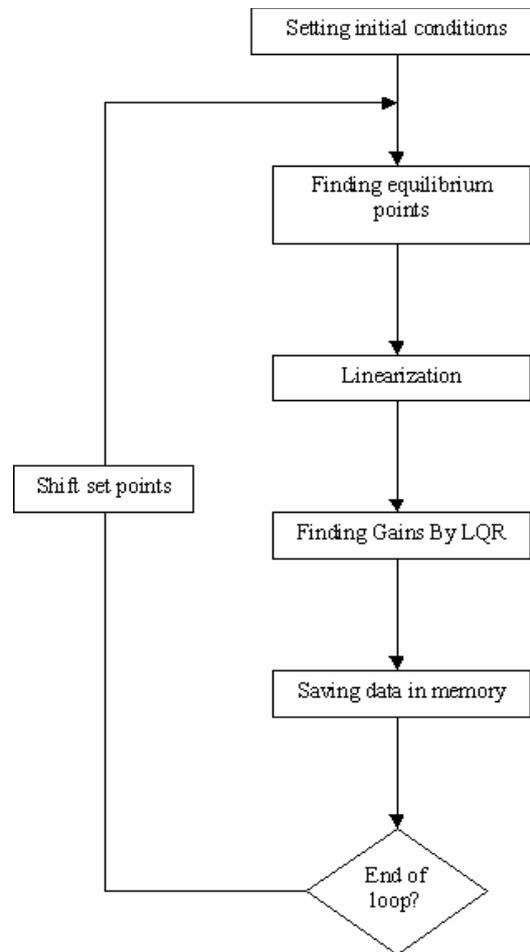


Figure 6. Program Structure

The program starts by initializing the set points and initial torque to be used for trim command in Matlab® to find the closest equilibrium point. This will provide us the torque shown in figure3. The obtained equilibrium points will be used for linearizing the model using trim function to produce the linear state space matrices in the neighborhood of equilibrium.

Then LQR gains are calculated using the adaptive structure matrix Q and the fixed controller effort R. In the step of saving data in memory, the following values are arranged in a matrix to be fed to the Simulink® simulation file in a regular time bases of 0.005 sec:

- Time vector
- 4 state feedback gain vectors. which are :

$$x1 = q1, x2 = \dot{q}1, x3 = q2, x4 = \dot{q}2$$

- 2 set points vectors for the actuated and free joints.

The whole process is repeated in order to scan the trajectory from $-\pi$ to $\pi/2$.

5. Discussion and Simulation results:

In order to observe the performance of the proposed control law based on continuous gain scheduling, we performed simulations on MATLAB® using SIMULINK®. We considered the system taking parameters $\theta_i : i \in [1,2,3,4,5]$ of a real Pendubot from Quanser®.

$\theta_1 = 0.0761, \theta_2 = 0.0662, \theta_3 = 0.0316, \theta_4 = 0.9790, \theta_5 = 0.3830$, The initial condition of the model was

$q_1 = 2.9671 \quad \dot{q}_1 = 0 \quad q_2 = -1.3963 \quad \dot{q}_2 = 0$

With the following state feedback gains used to balance the Pendubot about this point:

$K = [-7844.6 \quad -1080.9 \quad -7671.0 \quad -1018.4]$

Simulations show that our continuous gain scheduling controller is forcing the pendulum to track smoothly the set points.

Figure 7 presents an overview of the Simulink® model used to perform the simulation: It has the following:

- 1- Pendulum Model: This model is a nonlinear model derived from equations (4),(5),(6),and (7).
- 2- Speed estimator, which differentiate the position readings and filtering it by taking the average of 3 sequential speed values.
- 3- Balance Controller, which receives the state variables and the updated set points $[y_1, y_2]$ and gains at x_3 .
- 4- System Update: which takes the updated set points and gains every time step $=0.05$ sec from the workspace in Matlab®.
- 5- Hold Buffer will allow the values to pass after 4 sec at which the updated values match the initial condition.

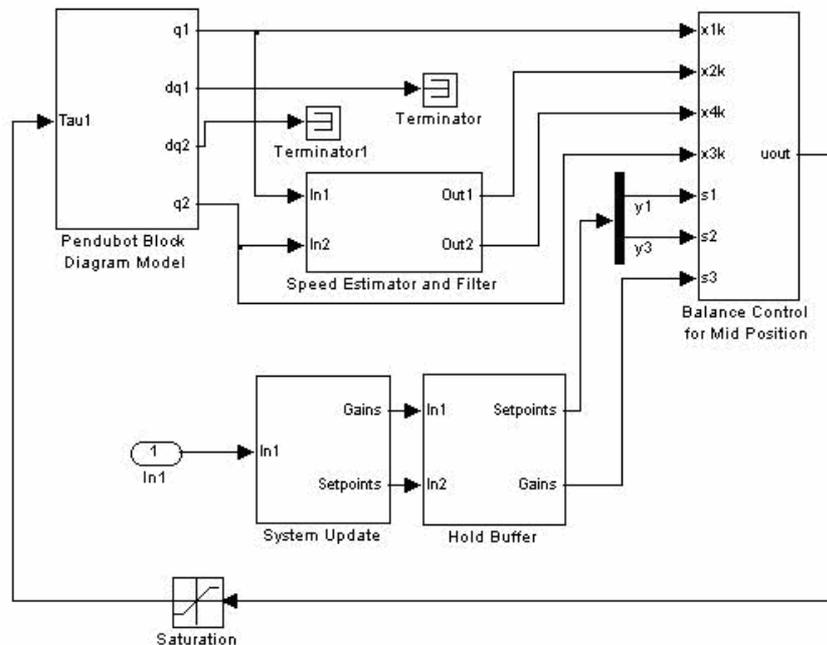


Figure 7. Simulink® Model

The simulation results show various simulation results of executing the file for 120 sec.

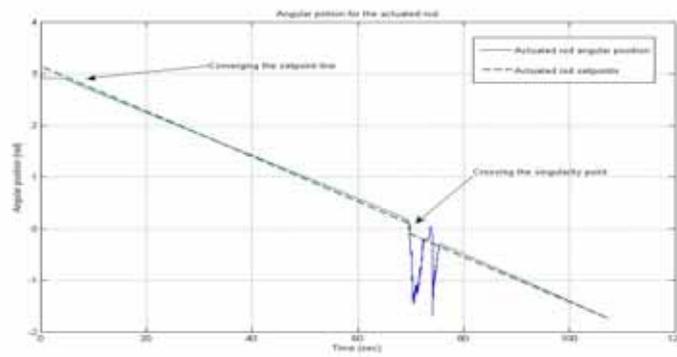


Figure 8. Actuated Joint Angular Position and its Set Point

From figure 8 we can see the sudden change in the set points from -0.1 to 0.1 rad and the effect of applying the open loop control at this neighborhood at around $t=70$ sec.

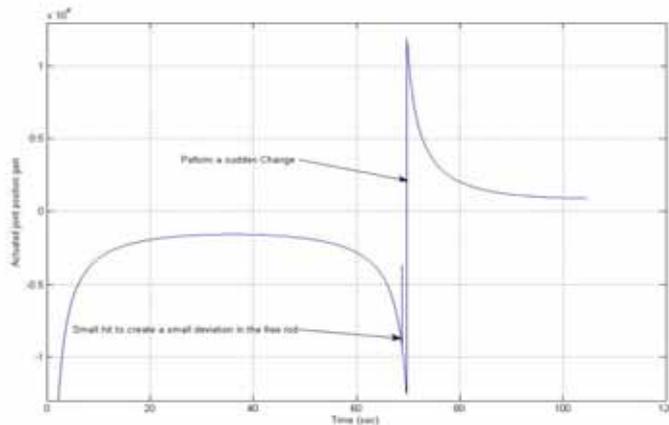


Figure 9. Gain k_1 Applied at The Actuated Joint Position State Value

Figure 9 shows the updated gain for x_1 and how it starts from a maximum value at the beginning and decreased to a minimum value when the pendulum approach the up right position where the control effort there is minimal. Also it shows the sudden spike made to create a small deviation in the free rod before passing the singularity envelope. This effect on the free joint angular position is shown in figure 10.

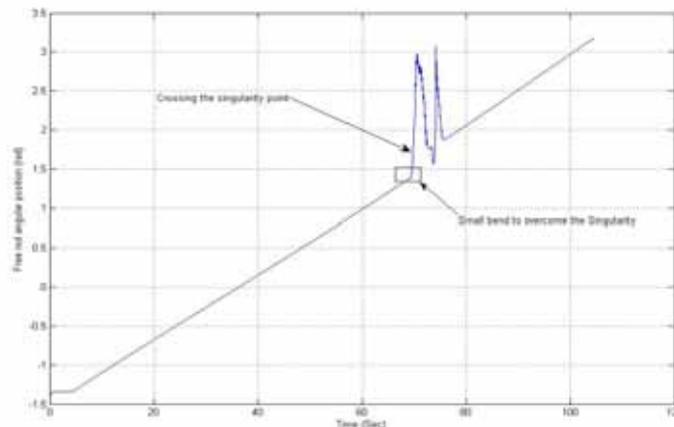


Figure 10. Free Joint Angular Position

Figure 11 shows the control action at the moment of passing the singularity region and how the spike in the gain shown before reflects on the control action. The large action coming after is due to the speed of the pendulum when we switched the controller back after passing the region. However this action can be decreased by making the singularity region narrower in order not to allow it to pass a long path which will give the chance to the speed to increase but on the other hand this will force us to use bigger gain values at the neighborhood of this region.

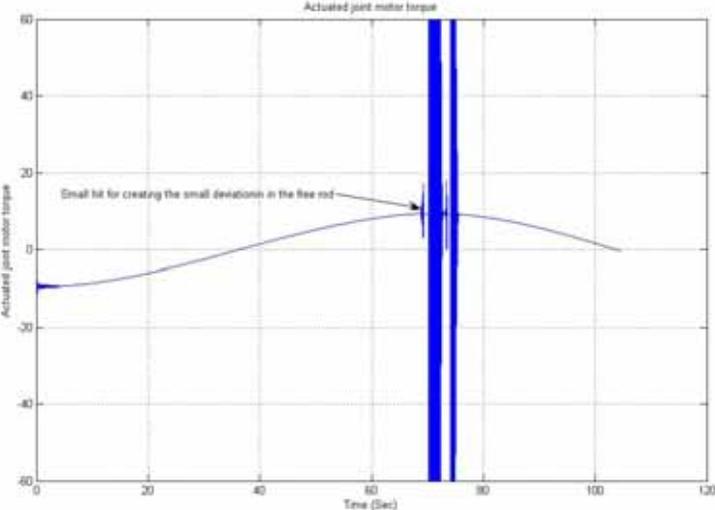


Figure 11. Actuated Joint Motor Torque

The velocity variation can be shown in figures 12 and 13 for both joints, the average of the angular speed outside the singularity region is about 0.04 rad/sec then it jumps to 1 rad/sec after passing this region which will cause a big action in the controller to overcome this big deviation from the set points.

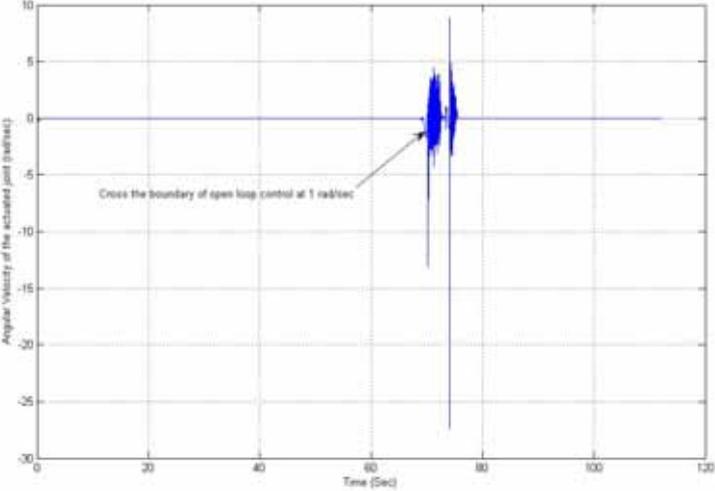


Figure 12. Actuated Joint Angular Velocity

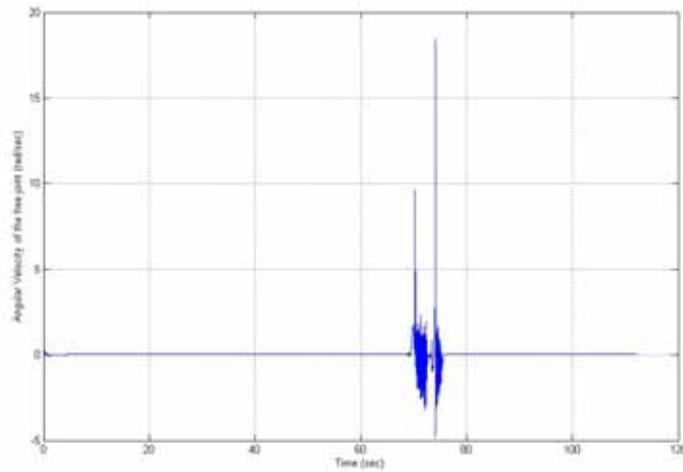


Figure 13. Free Joint Angular Velocity

Figure 14 shows the action of the switch controller and the time domain where the reduced action is taking place.

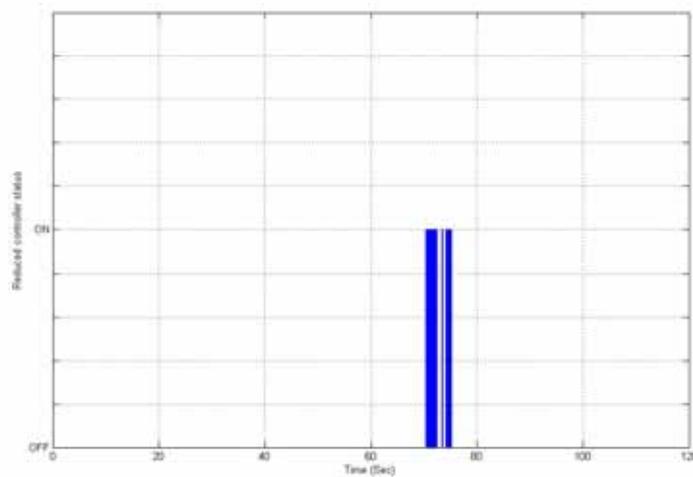


Figure 14 Switching Action

6. Conclusion:

We have presented a control strategy for the Pendubot that brings the pendulum from about a horizontal position to upper position and then pass the singularity region and complete its path to the mid position, we presented our algorithm and the design procedure for the continuous gain scheduling. Comparing with other methods presented, we were able to move the pendulum smoothly outside the singularity envelope. Passing the singularity region was also shown and how we control the system after passing this region to stabilize it back and track the set points smoothly. To the best of the authors' knowledge, this is the first implementation of a continuous gain scheduling to stabilize the Pendubot and bring it to cross singular points.

References:

- [1] Mark W.Spong. “*Underactuated Mechanical Systems*” Coordinated Science Laboratory, University of Illinois at Urbana-Champaign
- [2] Faical Mnif and Jawhar Ghommem, “*Genetic Algorithms Adaptive Control for an Underactuated System*”. International Journal of Computational Cognition, VOL. 3, NO. 1, March 2005.
- [3] Isabelle Fantoni, Rogelio Lozano, and Mark W. Spong. “*Energy Based Control of the Pendubot*”. IEEE Transactions on Automatic Control, Vol. 45, No. 4, April 2000
- [4] Mingjun Zhang, Tzyh-Jong Tarn. “*Hybrid Control of the Pendubot*”. IEEE/ASME Transactions on Mechatronics, VOL. 7, NO. 1, March 2002.
- [5] Wei Li, Kazuo Tanaka and Huo O.Wang. “*Acrobatic Control of a Pendubot*” IEEE Transactions on fuzzy systems, Vol. 12, NO. 4, August 2004.
- [6] Ofelia Begovich, Edgar N.Sanchez, Sanchez, Senior Member, IEEE, and Marcos Maldonado. “*Takagi-Sugeno Fuzzy Scheme For Real-Time Trajectory Tracking Of An Underactuated Robot*”. IEEE Transactions on control systems technology, Vol. 10, No. 1 January 2002.
- [7] Isabelle Fantoni, Rogelio Lozano, “*Non-linear Control for Underactuated Mechanical Systems*” ISBN 1-85233-423-1. Springer.
- [8] Mark W.Spong, Laurent Praly. “*Control of Underactuated Mechanical Systems Using Switching and Saturation*”. Coordinated Science Laboratory- University of Illinois and Centre Automatique et Systemes-Ecole des Mines de Paris.
- [9] G. Franklin, J. Powell, and A. Emami-Naeini. “*Feedback Control of Dynamic System*”. Addison Wesley, 1994.

The Fuzzy Inference System for Humanoid Robot Balance Control

Shu-Yun Chung⁺, Han-Pang Huang^{*}, Shu-Wen Yu⁺⁺

Robotics Laboratory, Department of Mechanical Engineering
National Taiwan University, Taipei, 10660, TAIWAN

^{*}Professor, ⁺ Graduate student, ⁺⁺ Undergraduate student

Email: hanpang@ntu.edu.tw

Abstract: *The balance control is the most important and fundamental part in biped robots. In traditional methods, a pre-planning trajectory which provides stable walking was off-line planned first. Then the trajectory will be modified by the real time feedback control. By this way, it can efficiently handle small disturbances. However, most of them fail for large disturbance. For human being, the step motion is commonly used to keep balance. But it requires fast and efficient computation to deal with falling down. To implement the step motion, a fuzzy inference system is proposed in this paper. Moreover, the step trajectory planning, which is different from walking trajectory, is also designed. The simulation shows excellent results to deal with large disturbance in different directions.*

1 Introduction

Due to numerous DOFs, the control of a humanoid robot is a difficult problem. Among different functions of humanoid robots, keeping balance is one of the essential abilities. In order to adapt to unknown environments, a fast and efficient balance control is necessary. However, the computation for balance control is huge and wastes lots of time due to its complicated nonlinearity. An intuitive approach is to utilize pre-planning trajectory for the required motions. The trajectory will be further modified in real time by perception system(Nagasaka, K., 1999). Although some well known criteria such as ZMP (zero moment point)(Sugihara, T., 2002), FRI (Foot rotation indicator)(Goswami, A., 1999) are very useful for achieving balance state, the realization with whole body cooperation is still complicated. The reason is that it is very difficult to define suitable postures which can regulate current unstable states. In other words, the key is how to determine exact corresponding postures to adjust humanoid robot stability.

Lots of papers(Yu Okumura, 2003)(Kajita, S., 2003,) were proposed to realize the real-time balance control system. But most of them were able to handle only small disturbances. Kudoh et al.(Kudoh, S.; 2002) utilizes quadratic programming and PD control

to accomplish the optimal balance motion in the 3D space, but the feet must touch the ground and cannot move. Namely, the upper limbs and waist need to operate in large motion for balance. It will require more DOFs to perform this action. In contrast to human, the step motion is commonly used for balance when we are falling down. It is the most efficient motion to relax the falling crisis.

The question is that it is difficult to describe the step motion because of the nonlinear system behavior. In order to analyze human balance problem, some simplified models, such as inverted pendulum, were proposed to describe the humanoid robot. However, it is still different from the real dynamic system. The modeling error is inevitable. Moreover, to achieve precise control will need large computation that is hard to be realized in real time. Compared with human, human learns the balance skill from experiences in daily lives, and keep balance without second thought. In order to imitate human balance skill, the fuzzy machine is proposed for the humanoid robot balance control in this paper. Based on the fuzzy rules, we can easily describe suitable motion to avoid falling down.

In addition, most people seldom notice that the step motion and walking motion are different. Since the step motion appears in emergency, it always requires a shorter trajectory to complete the step motion. On the other hand, the walking trajectory mainly focuses on stability. It is not suitable for describing the step motion. We will discuss and analyze the step trajectory in this paper. Moreover, a new step trajectory concept will be proposed. The step motion is an efficient strategy to stabilize the inclined state. In this paper, we only analyze the step motion in the standing state. However, we believe that the proposed fuzzy control concept is still effective for walking states.

This paper is organized as follows: the entire process will be described in section 2. In this section, we will briefly explain each stage of the balance process and give an overview of the paper. The major part of this paper is the fuzzy inference system and will be introduced in section 3. The fuzzy inference system will be integrated into the balance control system to determine step parameters. In section 4, the method to produce the step trajectory will be stated. In section 5, the quadratic programming method is applied to evaluate the variation of joint angles with different motions. Finally, the simulation results will be shown in section 6. Some assumptions and environment parameters will be further explained in section 6.

2 The Overview of The Process

The entire balance process is shown in Fig. 1. The falling down action is detected by a tilt angle. When the tilt angle is larger than a threshold, the robot will execute the corresponding step motion. Next, the stepping leg is determined by COM (center of mass) position. Like human, when we are falling down, the support leg will be the one which is close to COM.

Then, we move another leg to perform the stepping motion. By this way, the stepping leg can be easily identified.

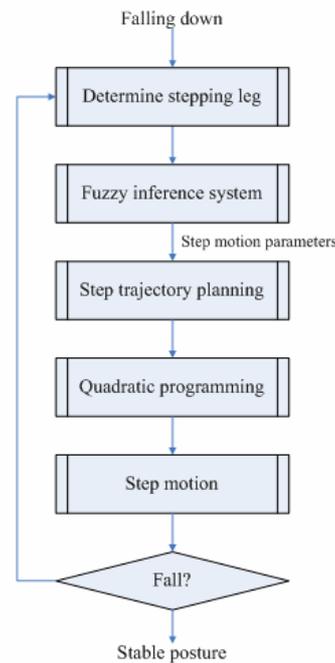


Fig. 1 The entire process

In the next stage, the step motion parameters are produced by the fuzzy inference system. The input of fuzzy inference system consists of the falling direction and velocity of the robot. With the information, the fuzzy machine will produce the step distance, direction, and velocity. The detail will be described in section 3.

According to those step parameters, it is easy to generate step trajectory. Through the quadratic programming method, each joint angle of the limbs can be obtained. Then the robot will execute the step motion to achieve balance state. If the robot does not keep balance after the step motion, another step motion will be performed until it reaches stable posture. In general, less than three steps are enough to overcome most disturbances. In the following sections, those stages will be discussed in detail.

3 Fuzzy Inference System

The fuzzy inference system is the kernel of the balance system. It produces various step parameters with different falling velocities and directions. In other words, the humanoid robot can dynamically adjust corresponding step motion until it reaches stable posture.

The inputs of the fuzzy inference system include falling velocity and direction. The membership functions are shown in Fig. 2. The falling direction is divided into eight

partitions. Fig. 3 gives their corresponding directions.

The outputs of the fuzzy system are step distance, step velocity and step direction. The membership functions are shown in Fig. 4. Because of the interference with legs, several things need to be noticed. If the stepping leg is left leg and the step direction is in the right hand side, it means the robot needs to cross its legs to complete the step motion. But most humanoids robot can not perform this kind of motion. In order to overcome this difficulty, the region of FRR and BRR will be replaced by FFR and BBR. Similarly, the region of FLL and BLL are replaced by FFL and BBL. As long as the stepping leg is in the same side as the step direction, the above problem will not occur. Then those limited areas can be selected. These actions can be accomplished by the fuzzy rules easily. The fuzzy rule is represented by

IF (fall direction is A) and (fall velocity is B)
THEN (the step distance is C) and (velocity is D) and (direction is E)

Finally, the COA (center of area) is used for defuzzification, and the fuzzy values will be transformed into crisp values.

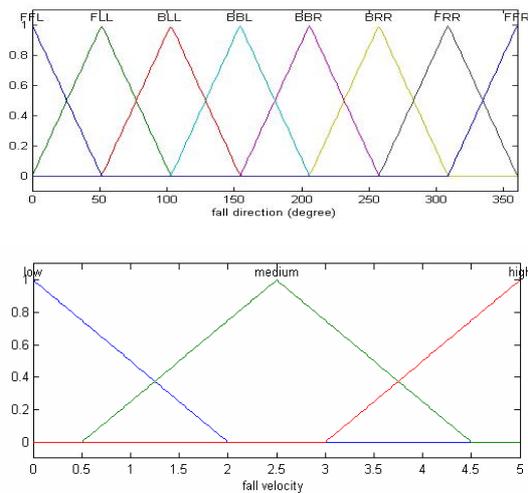


Fig. 2 Fuzzy membership function for fall direction and fall velocity

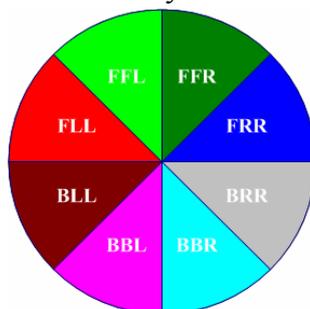


Fig. 3 The falling direction is composed of eight

directions (F: Front, B: Back, R: Right, L: Left)

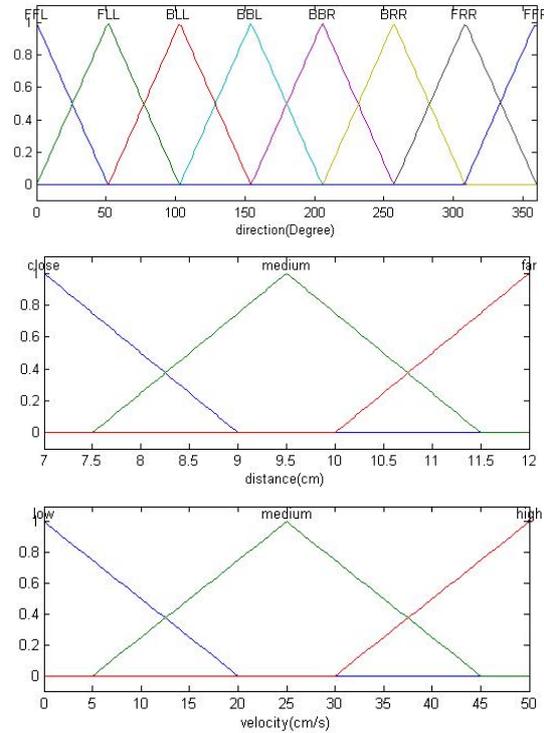


Fig. 4 fuzzy membership function for step direction, distance and velocity.

4 Step Trajectory Planning

To avoid falling down, the robot should be able to adjust its step trajectory dynamically with the unpredictable external force. Many methods of generating foot trajectories have been studied, such as recording human walking data(M. Y. Zarrugh, and C.W. Radcliffe, 1979), minimizing energy consumption (L. Roussel, and C. Canudas-de-Witi, 1998), and ZMP stability. Those methods mainly concerned about robot walking gait, but few methods discussed the step trajectory under the falling condition. In this paper, we focus on the analysis of the step trajectory with large impact.

The differences between the step trajectory and the walking trajectory are described below.

- 1) In the walking condition, the gait of the robot focuses on the stability and the height of the walking step. The highest point will be close to the position of the supported leg, as shown in Fig. 5. For the falling step trajectory, it requires a fast moving strategy. The robot should move its feet much farther per unit time. In other words, the highest point will be farther from the supported leg, as shown in Fig. 6.
- 2) Since the period of falling down is very short, the robot should not take much time to lift

its feet. To extend the distance of stride as far as possible, the highest point should be small enough so that the robot can take a bigger stride. Thus the parameters of the trajectory will be different from those in walking. The detail will be discussed in the following paragraphs.

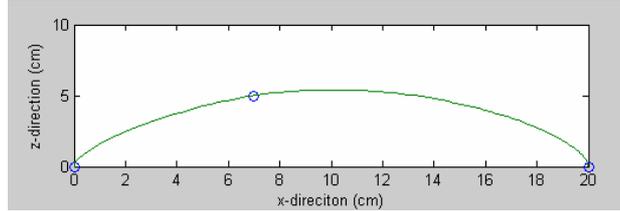


Fig. 5 General walking trajectory

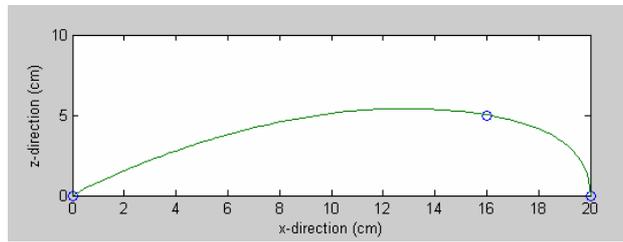


Fig. 6 Falling step trajectory

Many fitting algorithms can be used to smooth the trajectory, for instance, polynomial interpolation, linear interpolation, and cubic spline interpolation. Here, the cubic spline interpolation is selected due to its smoothness and simplicity.

The stepping parameters are shown in Fig. 7, where D_s is the length of one step; (L_{ao}, H_{ao}) is the highest point at time T_m ; T_c is the time when the foot contacts with the ground. Then the key knots of the cubic spline are defined as:

$$x_a(t) = \begin{cases} 0, & t = 0 \\ L_{ao}(D_s(k)), & t = T_m \\ D_s(k), & t = T_c \end{cases} \quad (1)$$

$$z_a(t) = \begin{cases} l_{an}, & t = 0 \\ H_{ao}(D_s(k)), & t = T_m \\ l_{an}, & t = T_c \end{cases} \quad (2)$$

where l_{an} is the length between the ankle and the foot.

From equations (1) and (2), (L_{ao}, H_{ao}) and T_m are function of D_s (equation (3)). Clearly, the position and time when the robot's foot reaches the highest point are dependent on the length of the step. These functions are described as:

ground during the stepping motion. Even though the robot is interrupted during the step motion, the stepping leg can still stand on the ground. Then the robot will try to keep balance again in the next step. Based on the above equations, the robot foot will follow the stepping trajectory.

6 Simulation

In order to justify the validity of the proposed algorithm, we use KHR-1 humanoid robot as the biped model. The kinematic structure, size, and mass are shown in Fig. 8. It has total 17 degrees of freedom. In detail, there are five DOFs in each lower limb, two in ankle, one in knee, and two in hip. Each arm has three DOFs, and head has one DOF.

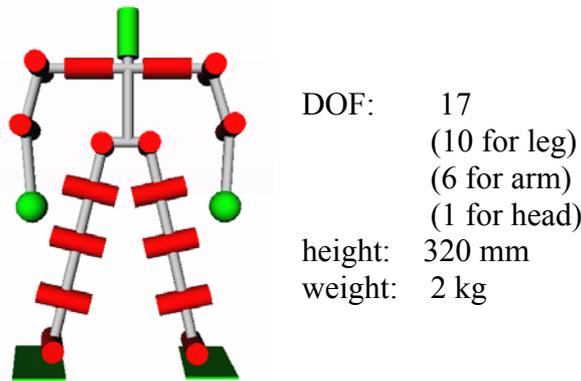


Fig. 8 The kinematic structure, size, and mass of KHR-1

To perform the falling motion, the inverted pendulum is adopted to simulate the real world situation. As shown in Fig. 9, the point mass represents the center of mass (COM) of the robot. We assume that the COM falling trajectory is the same as the trajectory of the mass of the inverted pendulum. In general, the inverted pendulum will perform stable state if there is no external force acting on it. When the unstable condition occurs, the angular acceleration of the inverted pendulum can be written as:

$$\alpha = \frac{g}{l} \sin \theta \quad (5)$$

where α is the angular acceleration. g and l are gravity and stick length, respectively. We rewrite equation (5) as

$$\begin{aligned} \dot{y}_1 &= y_2 \\ \dot{y}_2 &= \frac{g}{l} \sin(y_1) \end{aligned} \quad (6)$$

where y_1 and y_2 represent angle and angular velocity, respectively. Given initial condition (i.e. $y_1(0)$ and $y_2(0)$), we can easily obtain the falling trajectory. Here, initial angle velocity depends on the value of the external force. In addition, it should be noticed that the stick length is not always constant because of different COM positions. For example, the stick length should be shorter after the robot strides. Thus, we update the stick length every step.

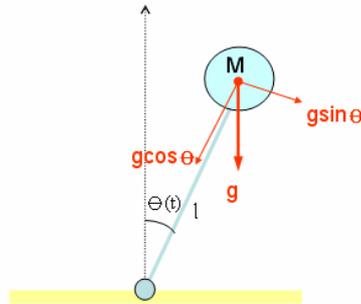


Fig. 9 The inverted pendulum model

Fig. 10~Fig. 13 show the simulation results under different external forces. Moreover, the real human actions are also presented in the experiment results to compare with the robot actions. However, the workspace of the robot is limited by the hardware structure. There are several actions, such as Fig. 12, that the real robot can not perform. The multi-step motion will be the solution of this interference problem. The robot does not need to cross its leg and still keeps balance with one more steps. The result is shown in Fig. 13.

7 Conclusions

In this paper, a fuzzy inference system is successfully applied to the humanoid robot balance control. Moreover, the falling step trajectory planning is discussed in this paper and the new concept of the step trajectory, which is different from the walking trajectory, is proposed. Further, the multi-step action is also realized to deal with large disturbance. Finally, the simulation shows excellent results and acts like real human.

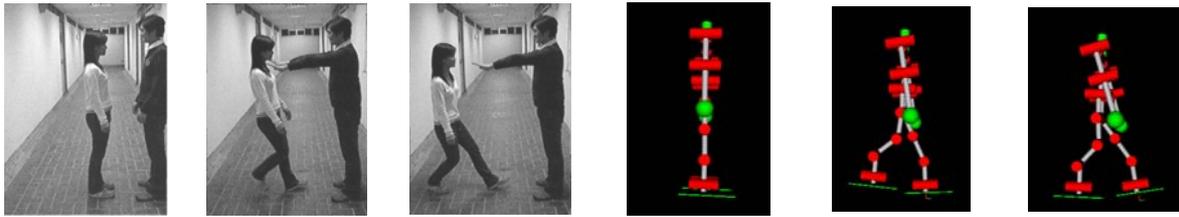


Fig. 10 The force is applied to the real human and the robot from the forward direction.

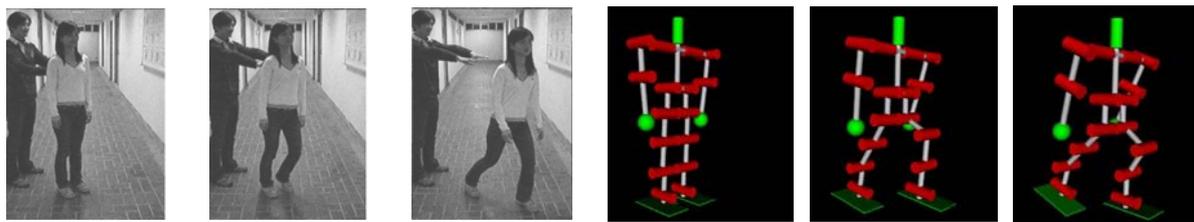


Fig. 11 The force is applied to the real human and the robot from the backward direction

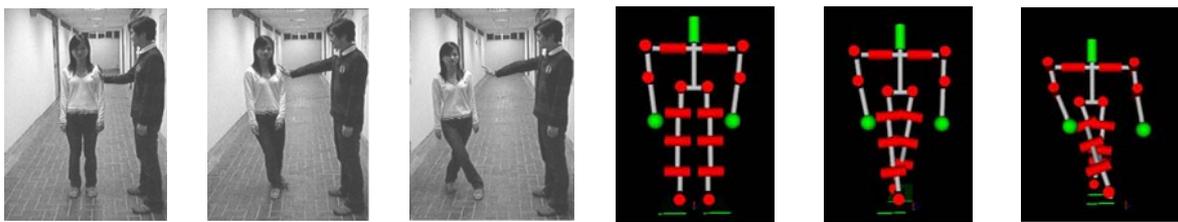


Fig. 12 The force is applied to the real human and the robot from the leftward direction

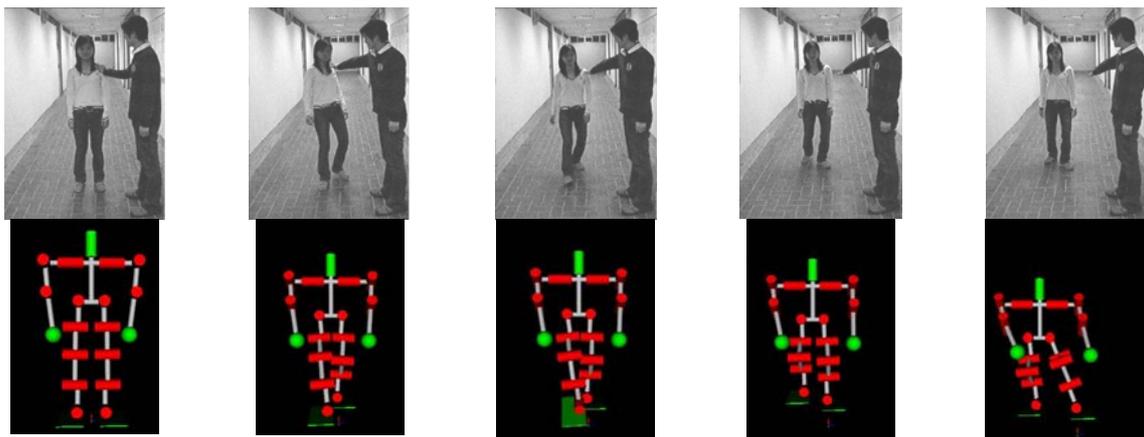


Fig. 13 The real human and the robot perform two steps to keep balance after suffering a large disturbance.

Reference

- Goswami, A., 1999, Foot rotation indicator (FRI) point: a new gait planning tool to evaluate postural stability of biped robots, IEEE/ICRA, 1999.
- Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H., 2003, Biped walking pattern generation by using preview control of zero-moment point, IEEE/ICRA, 2003.
- Kudoh, S.; Komura, T.; and Ikeuchi, K.; 2002, The dynamic postural adjustment with the quadratic programming method, IEEE/RSJ, 2002.
- L. Roussel, C. Canudas-de-Wit, and A. Goswami, 1998, Generation of energy optimal complete gait cycles for biped robots, IEEE/ICRA, 1998.
- M. Y. Zarrugh, and C.W. Radcliffe, 1979, Computer generation of human gait kinematics, J. Biomech., 1979.
- Nagasaka, K., Inoue, H., and Inaba, M., 1999, Dynamic walking Pattern Generation for a Humanoid Robot Based on Optimal Gradient Method, IEEE/SMC, 1999.
- Sugihara, T., and Nakamura, Y., 2002, Whole-body cooperative balancing of humanoid robot using COG Jacobian., IEEE/RSJ, 2002
- Yu Okumura, Tawara, T., Endo, K., Furata, T. Shimizu, M., 2003, Realtime zmp compensation for biped walking robot using adaptive inertia force control, IEEE/RSJ, 2003.

Control design for differential steering robot

Gregor Klančar, Drago Matko

University of Ljubljana, Faculty of Electrical Engineering

Tržaška 25, SI-1000 Ljubljana, Slovenia

email: gregor.klancar@fe.uni-lj.si, drago.matko@fe.uni-lj.si

Received May 25, 2005

Abstract

This paper presents a path tracking algorithm for nonholonomic mobile robot with differential drive. On the basis of robot kinematics equations a robot control is designed where the robot is controlled to follow the arbitrary path reference with a predefined velocity profile. For the control algorithm a system stability is proven in the sense of Lyapunov. The designed control algorithm proved stable and robust to the errors in robot initial positions, to input and output noises and to other disturbances. An example demonstrating control law operation on a simple trajectory example is also shown.

1 Introduction

The mobile robot control on a reference path is a well known problem that has been studied by many authors (Kolmanovsky, I., and McClamroch, N.H., 1995, Laumond, J.P., 1998, Luca, A., and Oriolo, G., 1995, Sarkar, N., *et al.*, 1994). Nonholonomic systems have motion limitations emerging from their kinematics model. Therefore some directions of motion are not possible. This paper deals with a mobile robot with differential drive which cannot move in the direction lateral to the wheels. The control of this robot is solved by considering its first order kinematics model. The obtained motion can later be upgraded to include the dynamics properties also (inertia and mass) where the robot control problem is transferred to a control of a second order kinematics model (Luca, A., and Oriolo, G., 1995, Sarkar, N., *et al.*, 1994). The control of mobile robots considering only first order kinematics is very common in literature (Canudas de Wit, C., and Sordalen, O. J., 1992, Laumond, J.P., 1998, Oriolo, G., *et al.*, 2002, Balluchi, A., *et al.*, 1996) as well as in practice. This is mainly because the control problem is easier to solve, the system dynamics can usually be neglected (fast and strong motors), especially at moderate speeds and because the robot design sometimes does not allow torque or acceleration to be forced at robot input (only reference speed). The basic control of a mobile robot in obstacle-free environment can be solved by point-to-point control with classic control where the intermediate course of the states between start and end state is not important. The other possibility is to control the robot to follow the reference

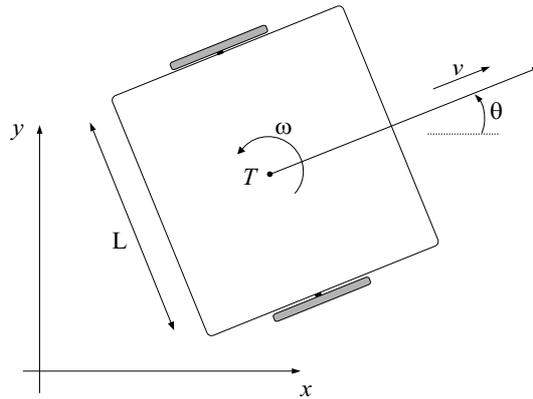


Figure 1: Robot architecture and symbols.

path from the start to the end position. When controlling nonholonomic systems, it is usually more appropriate to control the system (by feedback) to follow the reference path (Luca, A., and Oriolo, G., 1995, Sarkar, N., *et al.*, 1994). When controlling the robot to the end point (classic control), the control law usually has to be discontinuous and time variable (Canudas de Wit, C., and Sordalen, O. J., 1992, Kolmanovsky, I., and McClamroch, N.H., 1995, Luca, A., and Oriolo, G., 1995). Further on the robot has to consider nonholonomic constraints so its path cannot be arbitrary. The robot usually moves in the environment with obstacles, limitations and other demands which all somehow defines desired robot path. All these facts give advantage to control on a reference path which should follow all kinematic constraints. Nonholonomic systems usually have feedforward control (Canudas de Wit, C., and Sordalen, O. J., 1992, Sarkar, N., *et al.*, 1994, Luca, A., and Oriolo, G., 1995, Oriolo, G., *et al.*, 2002) where system inputs are calculated from the known trajectory and all kinematics constraints are implicitly considered by trajectory design. However, the use of the open-loop control only (just feedforward) is practically useless because it is not robust to errors in initial system states and other disturbances during operation. Closed loop is therefore added for practical use. The above mentioned combination (feedforward control) is intuitive and suitable for most nonholonomic mechanical systems. The paper is organised as follows. First the mobile robot motion modelling is revealed, followed by the control law derivation, the stability analysis and an example of the used control. The paper ends with conclusions.

2 Mobile Robot Modelling

In the sequel the direct and the inverse kinematics for the mobile robot with differential drive are determined. The robot's architecture together with its symbols is shown in Fig. 1 where it is supposed that its point of geometric centre T and the centre of gravity coincide.

The motion equations are as follows

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

where v and ω are tangential and angular velocities of the platform in Fig. 1. Right and left velocities of the robot wheels are then expressed as $v_R = v + \frac{\omega L}{2}$ and $v_L = v - \frac{\omega L}{2}$.

For a given reference trajectory $(x_r(t), y_r(t))$ defined in a time interval $t \in [0, T]$ the feedforward control law can be derived. From the obtained inverse kinematics the robot inputs are calculated which drive the robot on a desired path only if there are no disturbances and no initial state errors. The needed robot inputs, tangential velocity v_r and angular velocity ω_r are calculated from the reference path. The tangential velocity reads

$$v_r(t) = \pm \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \quad (2)$$

where the sign depends on the desired drive direction (+ for forward and – for reverse). The tangent angle of each point on the path is defined as

$$\theta_r(t) = \arctan2(\dot{y}_r(t), \dot{x}_r(t)) + k\pi \quad (3)$$

where $k=0,1$ defines the desired drive direction (0 for forward and 1 for reverse) and function $\arctan2$ is the inverse tangens function which returns the correct angles in all situations. By calculating the time derivative of Eq. (3) the robot angular velocity is obtained

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} = v_r(t)\kappa(t) \quad (4)$$

where $\kappa(t)$ is path curvature. By following relations from (1) to (4) and the defined reference robot path $\mathbf{q}_r(t)=[x_r(t), y_r(t), \theta_r(t)]^T$ robot inputs $v_r(t)$ and $\omega_r(t)$ are calculated. The necessary condition in the path design procedure is twice differentiable path and nonzero tangential velocity $v_r(t) \neq 0$. If for some time t tangential velocity is $v_r(t)=0$, the robot rotates at a fixed point with the angular velocity $\omega_r(t)$. Angle $\theta_r(t)$ cannot be determined from Eq. (2) and therefore $\theta_r(t)$ must be given explicitly.

3 Control Design

When the robot is controlled to drive on a reference path, it usually has some following error. The following error expressed in the frame of the real robot, as shown in Fig. 2, reads

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (5)$$

In Fig. 2 the reference robot is an imaginary robot which ideally follows the reference path. On the contrary the real robot (when compared to the reference robot) has

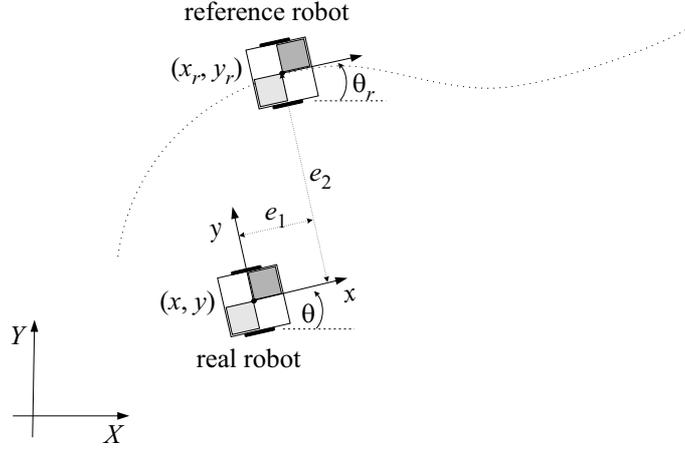


Figure 2: Robot following error transformation.

some error when following the reference path. Therefore the control algorithm should be designed to force the robot to follow the reference path precisely.

Considering the robot kinematics (1) and deriving relations (5) the following kinematics model is obtained

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{r1} \\ u_{r2} \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (6)$$

where u_{r1} is feedforward tangential velocity (2) and u_{r2} is feedforward angular velocity (4). Robot inputs, regarding relation (6), can be expressed in the following form

$$\begin{aligned} u_1 &= u_{r1} \cos e_3 - v_1 \\ u_2 &= u_{r2} - v_2 \end{aligned} \quad (7)$$

where the first parts on the right side of the equal sign are the feedforward inputs, while v_1 and v_2 are the inputs from the closed loop. From (7) expressing the closed-loop inputs and rewriting Eq. (6) results in

$$\begin{aligned} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} &= \begin{bmatrix} 0 & u_2 & 0 \\ -u_2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \\ &+ \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} \cdot u_{r1} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \end{aligned} \quad (8)$$

Further on linearizing (8) around the operating point OP (OP: $e_1 = e_2 = e_3 = 0$, $v_1 = v_2 = 0$) results in the following linear model

$$\Delta \dot{\mathbf{e}} = \begin{bmatrix} 0 & u_{r2} & 0 \\ -u_{r2} & 0 & u_{r1} \\ 0 & 0 & 0 \end{bmatrix} \cdot \Delta \mathbf{e} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \Delta \mathbf{v} \quad (9)$$

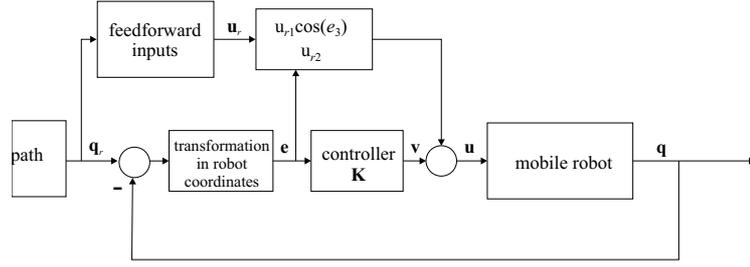


Figure 3: Mobile robot control schematic.

which is in the state space form $\Delta \dot{\mathbf{q}} = \mathbf{A} \cdot \Delta \mathbf{q} + \mathbf{B} \cdot \Delta \mathbf{u}$. Kalman controllability matrix has full rank $rank(\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B})=3$ if either u_{r1} or u_{r2} is nonzero which is sufficient condition for controllability only when reference inputs u_{r1} and u_{r2} are constant. However this is only true in case of linear and circular paths. The controllability of a driftless system can be ascertained by Chow's theorem if the system is completely nonholonomic (all constraints are nonholonomic). The system (1) is completely nonholonomic as it has only one nonholonomic constraint $y_c \cos \theta - x_c \sin \theta = 0$ (robot cannot move in lateral direction to its wheels) and therefore the system is controllable.

Let us further on define the linear state space controller for the closed loop. The system has three states and two inputs, thus

$$\mathbf{v} = \mathbf{K} \cdot \mathbf{e} \quad (10)$$

where gain matrix \mathbf{K} has dimensions 2×3 . The schematic of the obtained control is explained in Fig. 3.

The controller structure can be derived from observing Fig. 2. To reduce error in the driving direction e_1 the tangential robot velocity should be changed correspondingly. Similarly, the orientation error e_3 can be efficiently manipulated by robot angular speed. Finally, the error orthogonal to the driving direction could be reduced changing the angular velocity. At the same time also the robot drive direction (forward or backward) should be considered. From the above conclusions the control law is

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ 0 & -\text{sign}(u_{r1})k_2 & -k_3 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (11)$$

The next question is how to determine the suitable controller gains. This can always be done by trying or by optimization of some cost function. In this paper the controller gains are determined by comparing the real and the desired characteristic polynomials. The polynomial takes the following form

$$(s + 2\zeta\omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2) \quad (12)$$

Similarly as for the second order systems the desired damping coefficient $\zeta \in (0,1)$ and the characteristic frequency $\omega_n > 0$, are selected. An extra pole at $s = -2\zeta\omega_n$ increases the rising time and decreases the system overshoot. A characteristic polynomial of a closed

loop with a state space controller (11) is

$$\det(s\mathbf{I} - \mathbf{A} + \mathbf{BK}) = s^3 + (k_1 + k_3)s^2 + (k_1k_3 + k_2u_{r1} + u_{r2}^2)s + k_1k_2eu_{r1} + k_3u_{r2}^2 \quad (13)$$

Comparing coefficients at the same power of s in Eqs. (12) and (13) result

$$\begin{aligned} k_1 + k_3 &= 4\zeta\omega_n \\ k_1k_3 + k_2u_{r1} + u_{r2}^2 &= 4\zeta^2\omega_n^2 + \omega_n^2 \\ k_1k_2u_{r1} + k_3u_{r2}^2 &= 2\zeta\omega_n^3 \end{aligned} \quad (14)$$

Further on let us find the solution of equation (14) in the form suggested in (Luca, A., *et al.*, 2001)

$$k_1 = k_3 = 2\zeta\omega_n \quad (15)$$

k_2 is then determined as

$$k_2 = \frac{\omega_n^2 - u_{r2}^2}{|u_{r1}|} \quad (16)$$

The obtained system poles are constant

$$\begin{aligned} s_1 &= -2\zeta\omega_n \\ s_2 &= -\zeta\omega_n + \omega_n\sqrt{(\zeta^2 - 1)} \\ s_3 &= -\zeta\omega_n - \omega_n\sqrt{(\zeta^2 - 1)} \end{aligned} \quad (17)$$

and ω_n should be larger than the maximum allowed robot angular velocity, $\omega_n \geq u_{r2MAX}$. When u_{r1} is close to zero, k_2 goes to infinity and therefore a gain scheduling (Luca, A., and Oriolo, G., 1995) should be chosen for k_2 as $k_2 = g \cdot |u_{r1}(t)|$. System characteristic frequency becomes

$$\omega_n(t) = \sqrt{u_{r2}^2(t) + gu_{r1}^2(t)} \quad (18)$$

and the controller gains are

$$\begin{aligned} k_1 &= k_3 = 2\zeta\omega_n(t) \\ k_2 &= g \cdot |u_{r1}(t)| \end{aligned} \quad (19)$$

The parameter $g > 0$ gives an additional freedom in controller design (Luca, A., *et al.*, 2001). The controller gains approaches zero when the robot stops at which time the robot is not controllable. If the controller gains are chosen from Eqs. (15) and (16) a nonlinear, time varying controller is obtained which gives stable and constant poles of the closed-loop system. Although the poles are stable and constant, the local asymptotic stability is not guaranteed as system is still time varying (Luca, A., *et al.*, 2001, Oriolo, G., *et al.*, 2002). System stability is therefore checked by Lyapunov stability analysis discussed bellow.

4 Stability Analysis

For the presented controller (7) and (11), the system stability in the sense of Lyapunov is shown next. Lyapunov energy function $V(\mathbf{q})$ should be positive definite and continuous

with continuous derivatives on all states \mathbf{q} . In the equilibrium point $\mathbf{q}=\mathbf{0}$ the energy function must be $V(\mathbf{q}) = 0$.

The origin point $\mathbf{q}=\mathbf{0}$ is stable equilibrium point if there exists such a energy function $V(\mathbf{q})$ that $\dot{V}(\mathbf{q})$ is negative semidefinite over all the trajectory. If $\dot{V}(\mathbf{q})$ is negative definite, then the equilibrium point is asymptotically stable.

If the controller from (11) is rewritten considering (19), discontinuous part $k_2 \text{sign}(u_{r1})e_2$ is replaced by $\bar{k}_2 u_{r1} e_2$ where \bar{k}_2 is a positive constant ($\bar{k}_2 > 0$). The controller becomes

$$\begin{aligned} v_1 &= -k_1 e_1 \\ v_2 &= -\bar{k}_2 u_{r1} e_2 - k_3 e_3 \end{aligned} \quad (20)$$

Defining Lyapunov function as

$$V(\mathbf{e}) = \frac{1}{2} (e_1^2 + e_2^2 + e_3^2) \quad (21)$$

and its derivative as

$$\dot{V}(\mathbf{e}) = e_1 \dot{e}_1 + e_2 \dot{e}_2 + e_3 \dot{e}_3 \quad (22)$$

Considering model (9) and controller (20) the following relation is obtained

$$\dot{V}(\mathbf{e}) = -e_1^2 k_1 + e_2 e_3 u_{r1} - e_2 e_3 u_{r1} \bar{k}_2 - e_3^2 k_3 \quad (23)$$

from where the stability (negative semidefiniteness) cannot be concluded. A more appropriate energy function reads

$$V(\mathbf{e}) = \frac{\bar{k}_2}{2} (e_1^2 + e_2^2) + \frac{e_3^2}{2} \quad (24)$$

and its first derivative is

$$\dot{V}(\mathbf{e}) = -k_1 \bar{k}_2 e_1^2 - k_3 e_3^2 \quad (25)$$

which is negative semidefinite function (note that it is not negative definite with respect to e_2) and the origin $\mathbf{e} = \mathbf{0}$ is stable. Asymptotic stability is therefore not shown because $\dot{V}(\mathbf{e})$ is not negative definite. To check the asymptotic stability Barbalat's lemma is used. For energy function $V(t)$ holds

$$\begin{aligned} V(t) - V(0) &= \int_0^t \dot{V}(\tau) d\tau = \\ &= -k_1 \bar{k}_2 \int_0^t e_1^2(\tau) d\tau - k_3 \int_0^t e_3^2(\tau) d\tau \end{aligned} \quad (26)$$

The limit of the above integral as t tends to infinity is

$$\lim_{t \rightarrow \infty} \left(k_1 \bar{k}_2 \int_0^t e_1^2(\tau) d\tau + k_3 \int_0^t e_3^2(\tau) d\tau \right) \leq V(0) < \infty \quad (27)$$

The limit (27) exists and is a finite number; therefore $e_1, e_3 \in L_2$. Supposing the reference trajectory is smooth continuous function with bounded continuous derivatives then from (9) follows that u_{r1} and u_{r2} as well as \dot{e}_1, \dot{e}_2 and \dot{e}_3 are bounded. From $e_1, e_3 \in L_2$

and bounded derivatives \dot{e}_1 , \dot{e}_2 and \dot{e}_3 (using Barbalat's lemma) follows that $e_1 \rightarrow 0$ and $e_3 \rightarrow 0$ when $t \rightarrow \infty$. The derived control law from Eq. (20) forces e_1 and e_3 to zero while this is not guaranteed for e_2 . An example of such a situation is when the robot is in parallel with the reference line trajectory, $e_1 = e_3 = 0$, $e_2 \neq 0$ and the reference tangential speed $u_{r1}=0$. In this situation the controller (20) will not force e_2 to zero. However, since the $\dot{V}(\mathbf{e})$ is negative semidefinite, the error e_2 is always bounded. When the trajectory is designed so that $u_{r1}(t)$ is nonzero the controller (20) tries to lower the error e_2 .

5 Control Example

As already concluded the open-loop control only (feedforward) does not give satisfactory results because it is not robust to the initial state errors, noise and other disturbances during the operation. Fig. 4 shows two simulated examples of such control. In the first situation the robot has initial states error (it does not start with the correct orientation) and in the second situation the robot inputs and outputs are contaminated by additional noise at inputs (disturbances at command reception, quantization) and at outputs (sensors).

For successful control the closed loop must be implemented. Results of closed-loop control with feedforward control, according to the schematic in Fig. 3, are shown in Fig. 5.

In Fig. 5 the robot follows the reference path despite noise and initial state error in robot orientation. The controller successfully suppresses the initial state error and follows the reference path with satisfactory accuracy. Smaller errors during following are result of exaggerated added noise.

6 Conclusion

The control of a mobile robot along the reference curve is presented where the controller consists of two parts: the feedforward control and the closed-loop control. The former uses robot inverse kinematics to calculate the feedforward inputs from the reference curve, while the latter cancels the effects of noise, disturbances and initial state errors. For the control algorithm a system stability is proven in the sense of Lyapunov. The presented example as well as the real applications prove that a mobile robot can follow the desired reference path according to the prescribed velocity profile with a satisfactory accuracy.

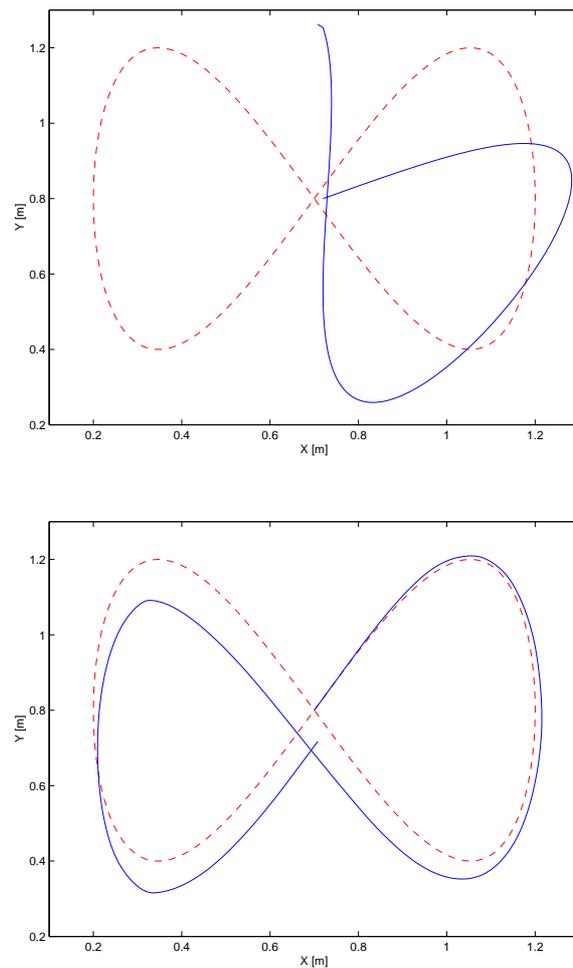


Figure 4: Open-loop control of a simulated robot (reference: - -, robot path: -). The first graph shows the robot with an initial state error and the second one the robot with added noise at its inputs and outputs.

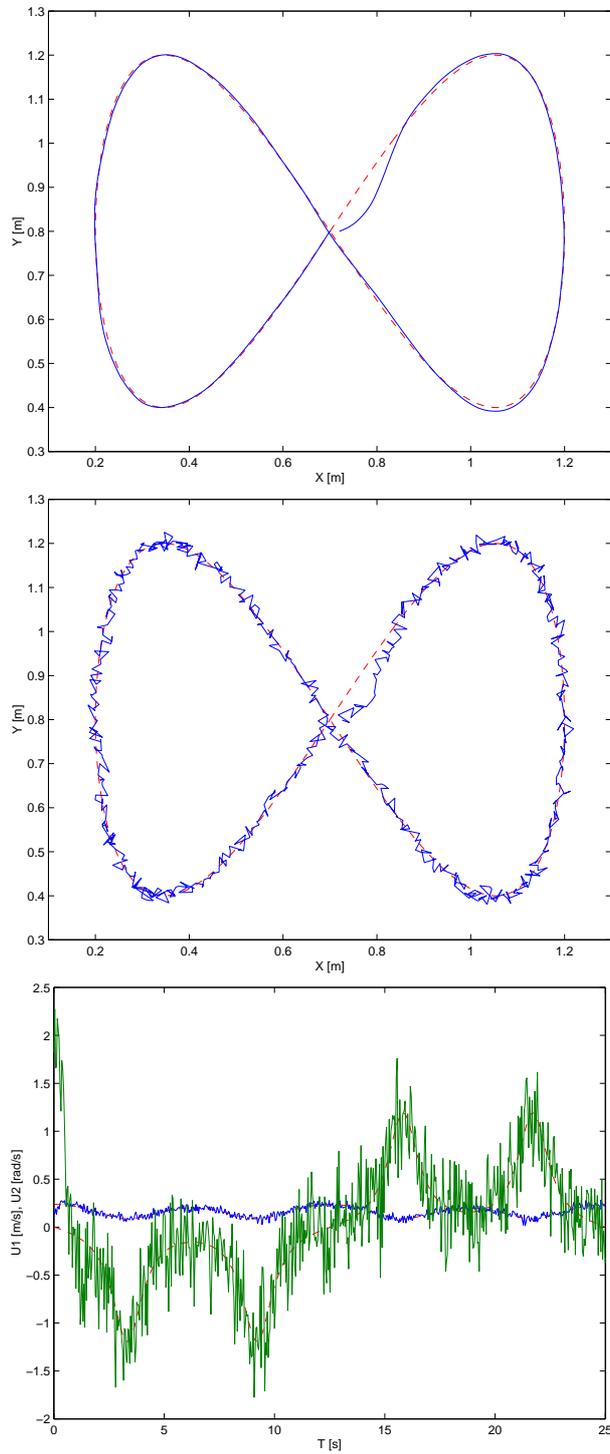


Figure 5: Closed-loop control ($\varepsilon = 0.6$, $\omega_n = 2$) of a simulated robot (reference: - -, robot path: -). The robot path, outputs with noise and inputs (feedforward: - -, feedforward and closed-loop: -).

References

- Balluchi, A., Bicchi, A., Balestrino, A., and Casalino, G., "Path Tracking Control for Dubin's Cars," in Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, pp. 3123-3128, 1996.
- Canudas de Wit, C., and Sordalen, O. J., "Exponential Stabilization of Mobile Robots with Nonholonomic Constraints," IEEE Transactions on Automatic Control, Vol. 37, No. 11, pp. 1791-1797, 1992.
- Kolmanovsky, I., and McClamroch, N. H., "Developments in Nonholonomic Control Problems," IEEE Control Systems, Vol. 15, No. 6, pp. 20-36, 1995.
- Laumond, J. P., Robot Motion Planning and Control, Lecture Notes in Control and Information Science, Springer-Verlag, 1998, vol. 229.
- Luca, A., and Oriolo, G., "Modelling and control of nonholonomic mechanical systems," in Kinematics and Dynamics of Multi-Body Systems, J. Angeles and A. Kecskemethy, Eds., Springer-Verlag, Wien, 1995.
- Luca, A., Oriolo, G., and Vendittelli, M., "Control of wheeled mobile robots: An experimental overview," in RAMSETE - Articulated and Mobile Robotics for Services and Technologies, Nicosia, S., Siciliano, B., Bicchi, A., and Valigi, P., Eds., Springer-Verlag, 2001.
- Oriolo, G., Luca, A., and Vandittelli, M., "WMR Control Via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation," IEEE Transactions on Control Systems Technology, Vol. 10, No. 6, pp. 835-852, 2002.
- Sarkar, N., Yun, X., and Kumar, V., "Control of mechanical systems with rolling constraints: Application to dynamic control of mobile robot," The International Journal of Robotic Research, Vol. 13, No. 1, pp. 55-69, 1994.

EVALUATION AND COMPARISON STUDY OF SHOOTING BALL ALGORITHMS

Alsaleem F M¹, and Al-Jarrah M A²
Mechatronics Center, American University of Sharjah

Abstract

This paper gives a survey on the current algorithms for shooting a ball in robot soccer. These methods will be compared in terms of their effectiveness. Simulation and experimental results of applying these algorithms to the AUS robot soccer team are compared.

A PD controller on the error between the current angle of the robot and the tangential angle of path- suggested by each method- with a constant linear velocity value is used to drive the robot. It will be shown that the dynamic behavior of the robot (slipping, and the minimum turn radius of the robot) have been neglected in these method. Neglecting dynamic characteristics can't be justified at high linear velocities.

1. Introduction

Robot soccer is an active area of research and it is the place where new and exciting technologies can be integrated and examined. Examples of technologies utilized are image processing, control theory, artificial intelligence, multi-agent systems, and motion planning and embedded systems [Werner, 2003]. Actually there are different types of schemes for soccer robot control available. In a remote brainless (vision-based), a host computer controls the robots by commanding the robot velocities. In the brain on-board (vision-based) soccer robot system, the robots move as per the vision data, keeping away from obstacles autonomously. In a robot-based scheme, each autonomous robot can make a decision based on information it collects with its sensors and, if needed can communicate each other [Dong-Han Kim,200]. This paper deals with the remote brainless robot system. Shooting a stationary ball into a given direction is the first step that one will consider before thinking about shooting a moving ball, which has many applications. From aerospace view the idea behind Robotic interception of moving objects can be treated as a problem of missiles tracking a flying target, also from industrial view they are many applications that used the same concept. Among the most efficient industrial application is APPE (Active Prediction, Planning and Execution) system for robot interception of moving objects. The key feature of the system is the ability of a robot to perform a task autonomously without complete *a priori* information. The APPE system has been tested for an object moving at a maximum velocity of 45mm/s [Messom.H, 2003].

¹AUS Mechatronics Graduate Students

¹AUS Mechatronics Director and Professor

2. LITERATURE REVIEW

Hitting a ball to a given target is a well known problem that has been studied by many authors, in [Peter,98] a low level learned multiagent system is introduced to shoot a ball in a simulator, they began their experimentation with the ball always being passed with the same trajectory and the same speed for all training and testing examples, then they varied the ball's speed keeping in mind that the whole trajectory of the ball is known, at the end they develop a learned system to handle the previous constraints together . New motion control system was proposed by CMUnited- 98 [Michae, 2000] for small-size league champion at RoboCup- 98. In this paper a new motion control algorithm is introduced, this algorithm allows a general differential-driven robot to accurately reach a target point with a desired orientation.

Geometrical solution for shooting a ball is presented in [Messom.H, 20033] and its shows how to use the kalman filter for: 1-reducing the noise that come from the analogue camera (odd and even scanning), and 2- to predict the future position of the moving ball. Potential field method [Dong-Han Kim, 200] introduced a way to shoot a fixed ball motion using potential field method.

3. KINEMATICS MODEL OF THE ROBOT:

A kinematics model of a two wheeled mobile robot with non-slipping wheels is considered [Dong-Han Kim, 200].

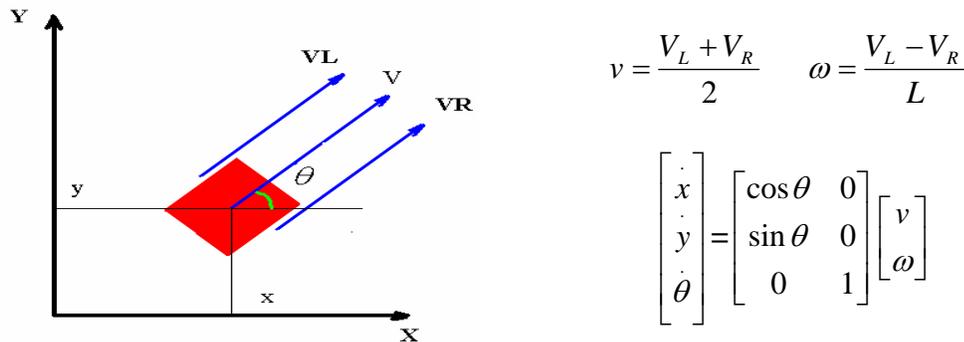


Figure 3.1: Robot Kinematics model

Equation in the right has three variables to be controlled, but only two inputs $[\omega \ v]$. This explains why, in general, no control is guaranteed to move the robot from a given posture (x, y, θ) to desired posture (x_d, y_d, θ_d) [Kim, 2004].

4. ALGORITHMS USED FOR KICKING A STATIONERY BALL

In this paper three algorithms are discussed and implemented on our AUS robot soccer:

1- The Geometrical Algorithm [Messom.H, 2003], 2-the CMUnited-98 shooting algorithm [Michael, 2000], and 3-the improved vector field method algorithm [Dong-Han Kim, 2000].

4.1- The Geometrical Algorithm

This algorithm calculates the left and right wheel velocities of the robot (vl, vr) to reach a desired position with a specific orientation:

$$vl = V_c - K\theta_e \quad \text{and} \quad vr = V_c + K\theta_e$$

Where:

K: is a proportional gain.

θ_e : is the angle between the required direction and the current robot direction as shown in Figure. 2.

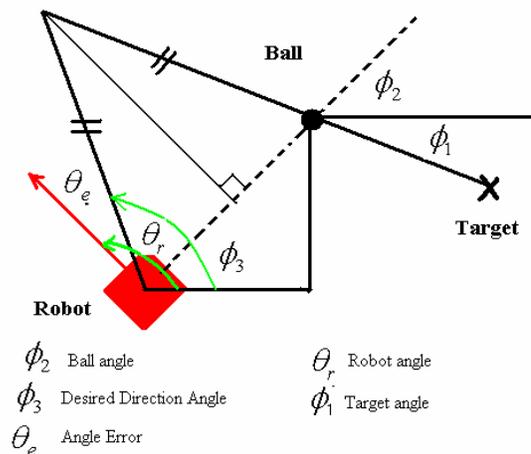


Figure 4.1: The geometrical method to shoot a fixed ball to a specific orientation

4.2- The CMUnited-98 shooting algorithm

Its gives reactive equations for deriving the left and right wheel velocities (vl, vr) in order to reach a target position:

$$vl = v(t - r) \tag{1}$$

$$vr = v(t + r) \tag{2}$$

Where:

$$(t, r) = \cos^2 \Delta \cdot \text{sgn}(\cos \Delta), \sin^2 \Delta \cdot \text{sgn}(\sin \Delta) \tag{3}$$

$$\Delta = \theta - \Phi \text{ (Figure: 3.a)} \tag{4}$$

These equations are extended for target configuration of the form (x^*, y^*, Φ^*) , where the goal; of the robot is to reach the target position (x^*, y^*) , while facing the direction (Φ^*) (Figure: 4.2.b).

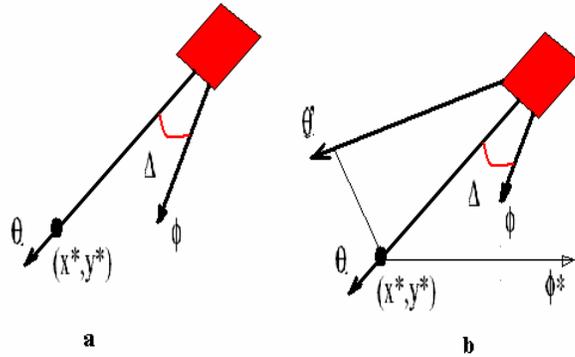


Figure 4.2: (a) - reaching the Target position without specific direction. (b) -Reaching the target position with a direction (Φ^*) .

4.3-The vector field method:

In Figure .4.3, the final position of the robot is the point G, and its final angle is to the right. The kick field at robot position P is given as:

$$\angle N(P) = \angle \vec{PG} - n(\angle \vec{PR} - \angle \vec{PG}) \quad (5)$$

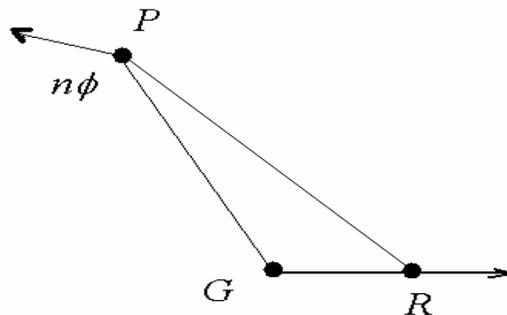


Figure 4.3: vector field method

5. IMPLEMENTATION OF THE SHOOTING METHODS ON AUS ROBOT SOCCER

For the implementation of all methods, the same formulas for the linear and angular velocity are used (Figure 5.1)

$$V = V_c \tag{6}$$

$$W = K_p (\angle \theta_{des} - \theta_{robot}) \tag{7}$$

Where:

V : is the linear velocity (constant value).

W : is the angular velocity.

θ_{des} : is the angle that it is calculated (According to the used method).

θ_{robot} : is the angle of the robot.

K_p : Proportional control gain.

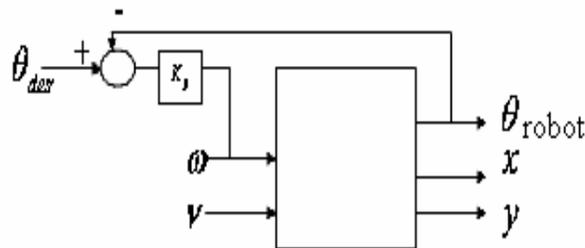


Figure 5.1: Block diagram of the control implementation of the three methods

5.1- How to determine the K_p value?

Determining the K_p value is not a straight forward because K_p is function of the linear velocity v . The minimum turn radius (r), that the robot can turn without slipping depends on v .

$$r = \frac{mv^2}{F_{cen}} \quad F_{cen} : \text{ centerfrugal force} \tag{8}$$

On the other hand, for a given value (v) the K_p gain will work fine until θ_e reaches a certain limits. This is significant when the (v) has a high value, for example Figure. 5.2-a shows the step response of the robot angle for 40° and 80° when $v=100\text{ cm/s}$ and $K_p=.5$, it can be seen that when the reference input is 40° the system is stable, but when the reference input is 80° the system is unstable. On the other hand Figure. 5.2-b, shows the step response of the robot angle for different step demands when $v=0\text{ cm/s}$ and $K_p=.5$. To improve the above algorithms the following modifications were tested:

- 1- Set the reference robot angle value at (40°).
- 2- Select the proportional gain for different values of v using the Zeigler Nichols closed-loop tuning method [Gene, 2002].

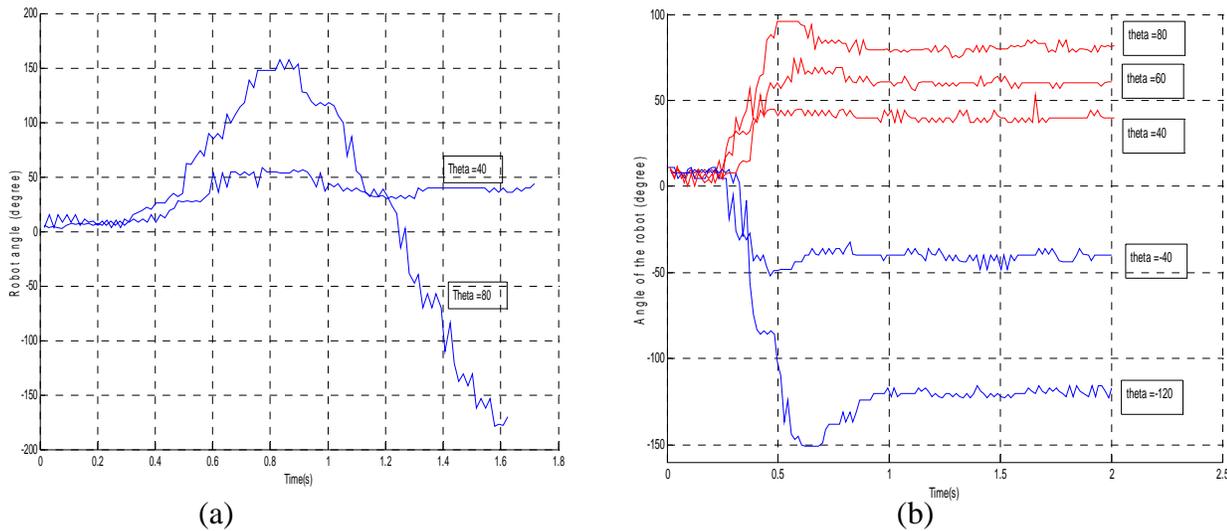


Figure 5.2: step response of the robot angle: a- $v=100\text{ cm/s}$, b- $v=0\text{ cm/s}$.

Applying these steps for AUS robot soccer for reference value of (40°) and for example $v=60\text{ cm/s}$, gives $K_u=.9$, so $K_p=.45$. Practical results of applying this value of K_p comparing with a result of applying K_p and K_d gains are shown in (Figure.5.3)

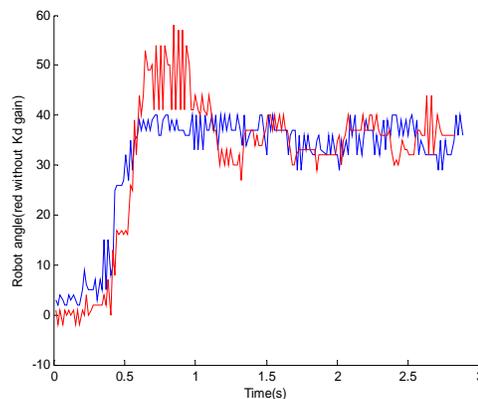


Figure 5.3: step response of the robot angle with PD controller

5.2- Experiments and Simulations Results

Now after finding the suitable gains of K_p for a certain values of v and at a fixed reference (40°), Practical and simulation results of the three mentioned methods are obtained using equations 6 and 7. The practical results are obtained by Applying the three methods on the AUS robot soccer players while the simulation results are obtained by Applying them on the Matlab simulation model of the robot.

5.2.1 The Geometrical Method results:

In this method a constant value of $v=120 \text{ cm/s}$ was used. To overcome the fact that the proportional control gain was designed based on reference angle step of 40° ; a condition in the code was introduced to reduce v whenever the error exceeds 40. Practical and simulation results are shown in (Figure.5.4).

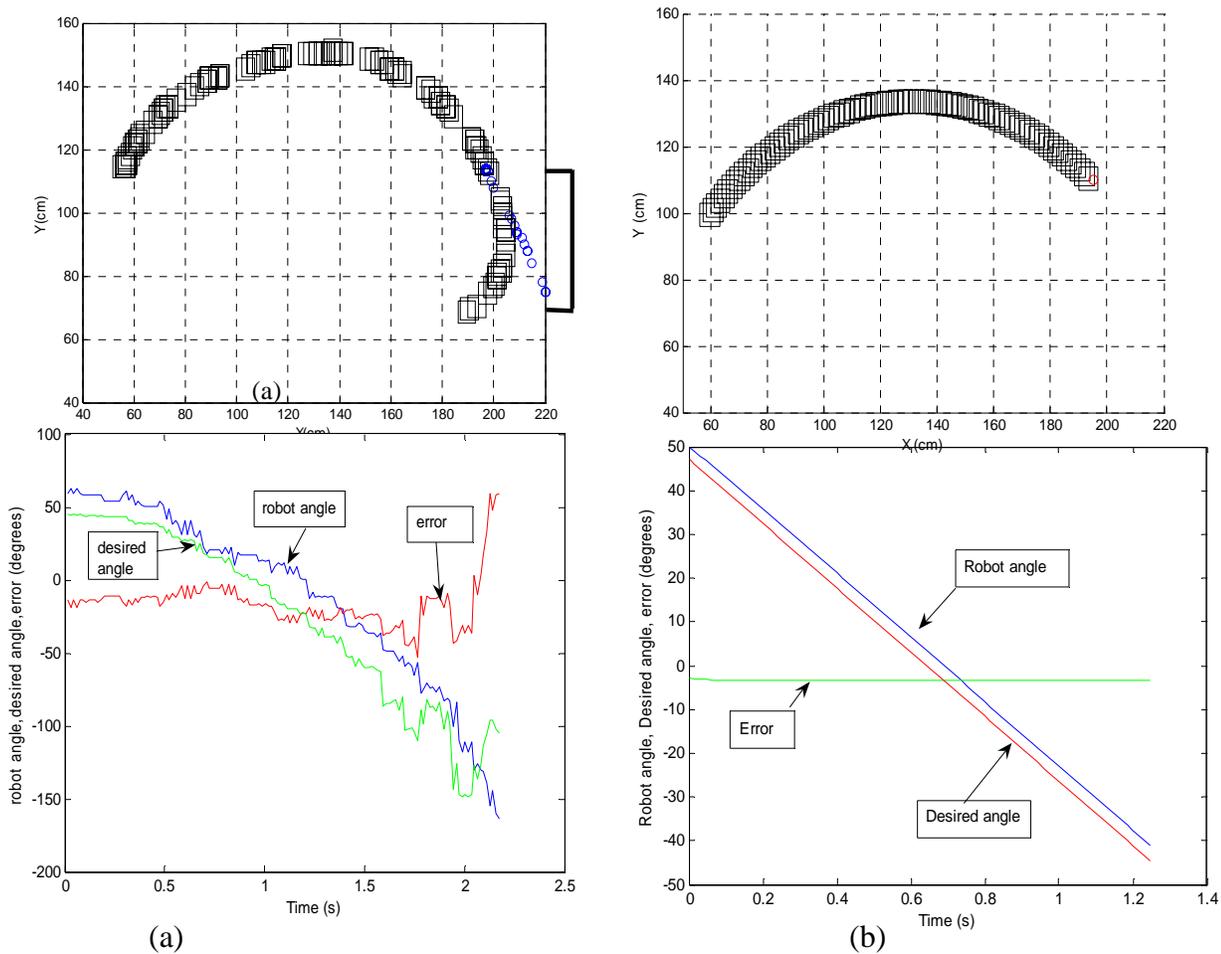


Figure.5.4: Experimental and simulation results of the geometrical method: a-experimental b-simulation

5.2.2 The CMU Method results:

A constant value of $v=100$ cm/s, was used, Practical and simulation results are shown in (Figure.5.5).

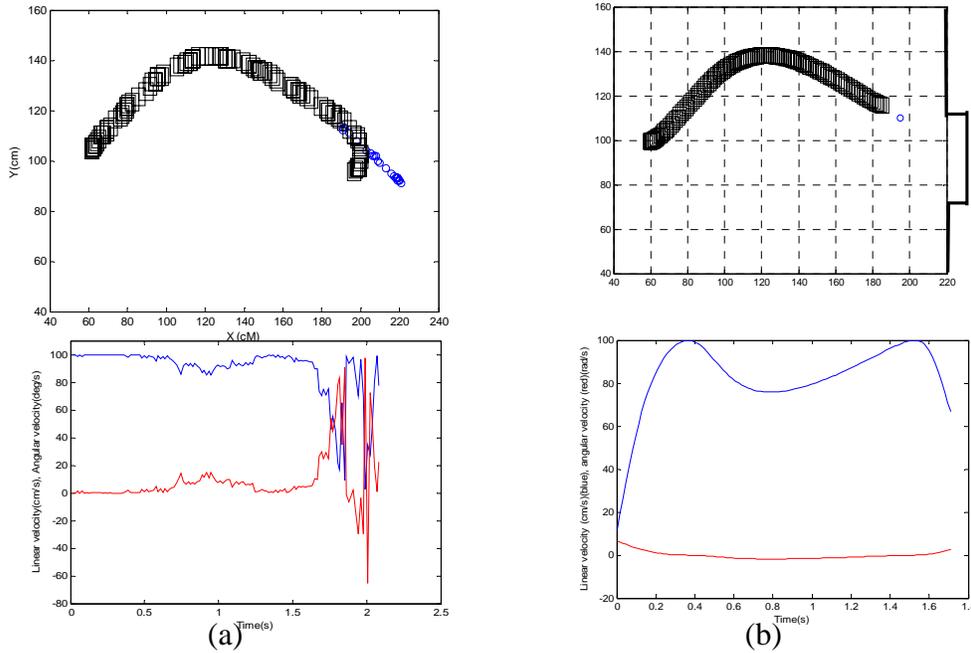


Figure.5.5: Experimental and simulation results of the CMU method: a-experimental b-simulation

5.2.2 The Potential field Method results:

A constant value of $v=80$ cm/s, was used, Practical and simulation results are shown in (Figure.5.6), It can be seen from the figures that: first, the path the robot follows is optimum- not like the path the robot follows in the two previous methods, and second, the robot after hitting the ball continue to push the ball until it enters the goal area 3- the ball exactly hit the center of the goal.

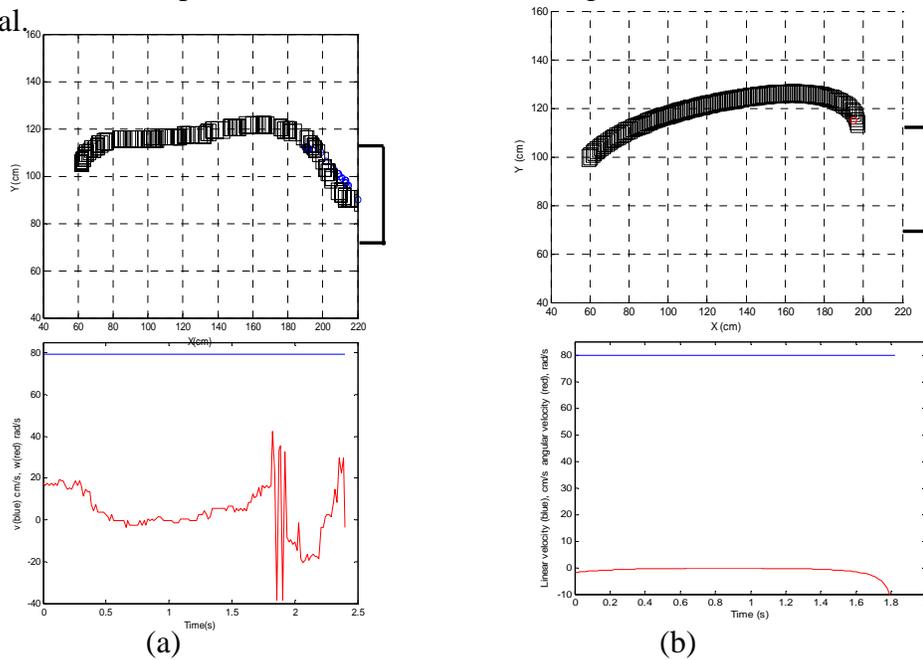


Figure.5.6: (a) Experimental and (b) simulation results of potential method

On other hand, unfortunately trying to increase the value of v up to 100 cm/s (see Figure.5.7), will make the system sometimes miss the ball).

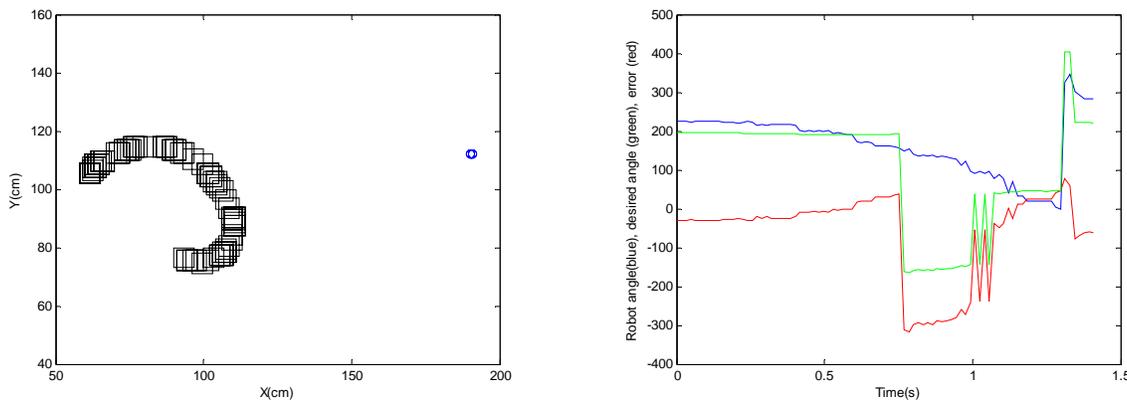


Figure.5.7: Experimental results of the Potential method: a-positions b- velocities experimental result $v=100$ cm/s.

5-CONCLUSION:

Three different methods to shoot a ball toward a given direction were surveyed in this paper; the geometrical, the CMUnited and the vector field methods. All these methods were simulated on Matlab before they were implemented on AUS robot soccer players using visual C++ language programming; it was shown that running all these methods at high speed will cause problems. Unless one considers the dynamic behavior of the robot, on the implementation, increased speed will be a problem. On the other hand, it was clear that the vector field method was the better method in term of optimizing the path between the robot and the ball, and in term of dripping the ball toward the center of the goal, but not in terms of running at a speed higher than 80 cm/s.

6-REFERENCES:

- [1] Dong-Han Kim, Yong-Jae Kim, Kwang-Choon Kim, and Jong-Hwan Kim, "Vector Field based Path Planning and Petri-net based Role Selection Mechanism with Q-learning for the Soccer Robot System", *Int. J. Intelligent Automation and Soft Computing*, Vol. 6, 2000, pp. 75-87
- [2] Gene F. Franklin, J. David Powell and Abbas Emami, "Feedback control of dynamic systems", 4th ed, Prentice Hall PTR, 2002.
- [3] Kim, J.-H., Kim, D.-H., Kim, Y.-J., and Seow, K.-T, "Soccer Robotics", Springer, 2004.
- [4] Messom1.C. H, G. Sen Gupta, S. Demidenko3 and Lim Yuen Siong, "Improving Predictive Control of a Mobile Robot: Application of Image Processing and Kalman Filtering", *Technology Conference Vail, CO, USA, May 2003*, pp.20-22.
- [5] Michael Bowling Manuela Veloso, "Motion Control in Dynamic Multi-Robot Environments", *Soccer World Cup III, 2000*, pp. 222–230.
- [6] Peter Stone and Manuela Veloso, "Towards Collaborative and Adversarial Learning: A Case Study in Robotic Soccer", Pittsburgh, *International Journal of Human Computer Studies*, 1998.
- [7] Werner. D. Dierssen "Motion Planning in a Robot Soccer System". Master's thesis, University of Twente, 2003.

System Identification Using a Modified Partial Recurrent Network

Muaiad Shawqi AL-Faysale, Senior IEEE Member

University of Technology
Department of Electrical and Electronic Engineering
P.O. Box 35010, Baghdad, Iraq
Email: msm@ieee.org

Abstract: *In this paper, a new structure for Partial Recurrent Neural Networks (PRNNs) is presented. The new structure is achieved by using adaptable interconnection weights between context layer nodes in addition to the adaptable links from hidden to context layer. This structure is very useful in solving many problems in the process of system identification and control of dynamical systems. The performance of this structure is compared with the performances of the Modified Elman Neural Network (MENN) and the Modified Recurrent Network (MRN). Simulation results obtained from the identification of linear and nonlinear dynamic systems showed that this structure gives less RMS error in minimum number of training cycles and minimum computational time.*

1 Introduction

Recurrent Neural Networks (RNNs) are an interesting class of models for time series processing, system identification and control problem. RNNs are able to build up internal memory suited for the task at hand and thus often lead to compact model representations. However, it is generally believed to be a difficult task to train this type of networks (Peddersen, M.W., 1997, Lin, T., Hene, B. et al, 1998).

The capacity of the internal memory of a RNN increases when the number of hidden units increases as it contains all information about previous inputs (Peddersen, M.W., 1997). In general, RNNs can be classified as fully and partially recurrent. Fully Recurrent Neural Networks (FRNNs) can have arbitrary feed-forward and feedback connections, all of which are trainable. In Partially Recurrent Neural Networks (PRNNs), the main network structure is feed-forward trainable connections. The feedback connections are formed through a set of "context" units and are not trainable. The context units memorize some past states of the hidden units, and so the outputs of the networks depend on an aggregate of the previous states and the current input. It is because of this property that PRNNs possess the characteristic of a dynamic memory (Pham, D.T., and Liu, X. L., 1997). The MENN is a special model of PRNNs. which have fixed gain in the self-feedback links of the context layer and unity gain in the feedback links from hidden layer to context layer (Pham, D.T., and Liu, X. L., 1997).

The MRN was proposed by (AL-Faysale, M.S.M., 2004 and 2005) has the same structure, shown in Fig. 1, of MENN but with modified training algorithm. MRN has two trainable weights of self-feedback of context layer and the feedback from hidden layer to context layer. In fact, PRNNs combine the advantages of both feed-forward and recurrent network (AL-Faysale, M.S.M., 2004). Many researchers (Sastry, B.S., et al., 1994, Santosand, J. and Duro, R.J., 1998, and Kwok, D.P., Wang, P., et al, 1994) paid a lot of attention to PRNN by applying it in control, identification, network generalization and signal processing.

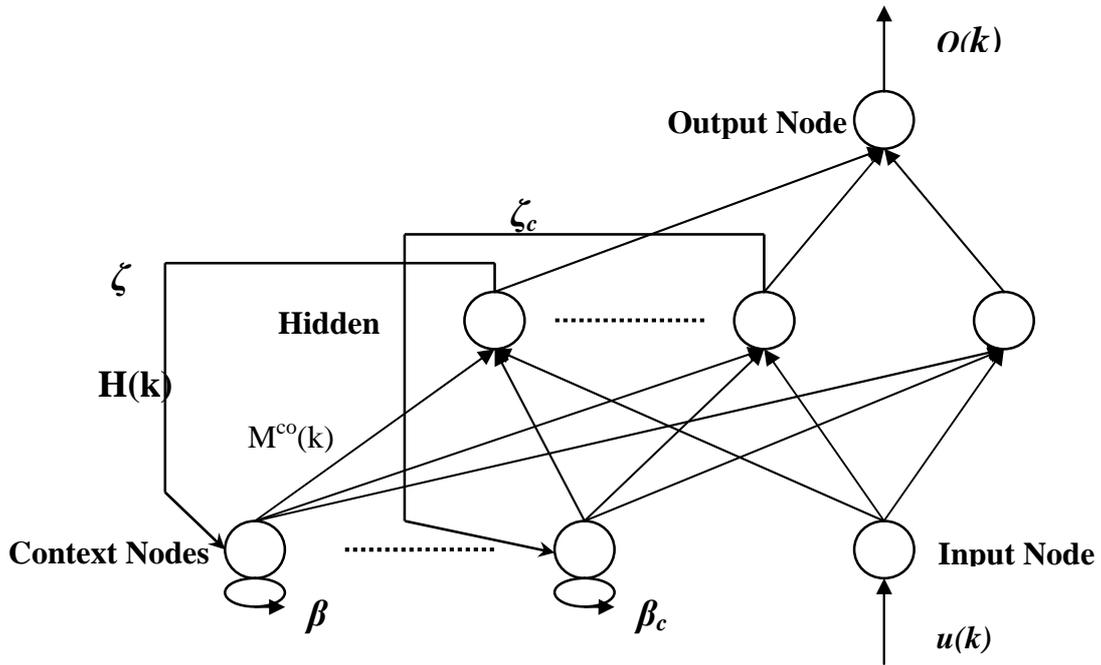


Fig. 1: The MRN.

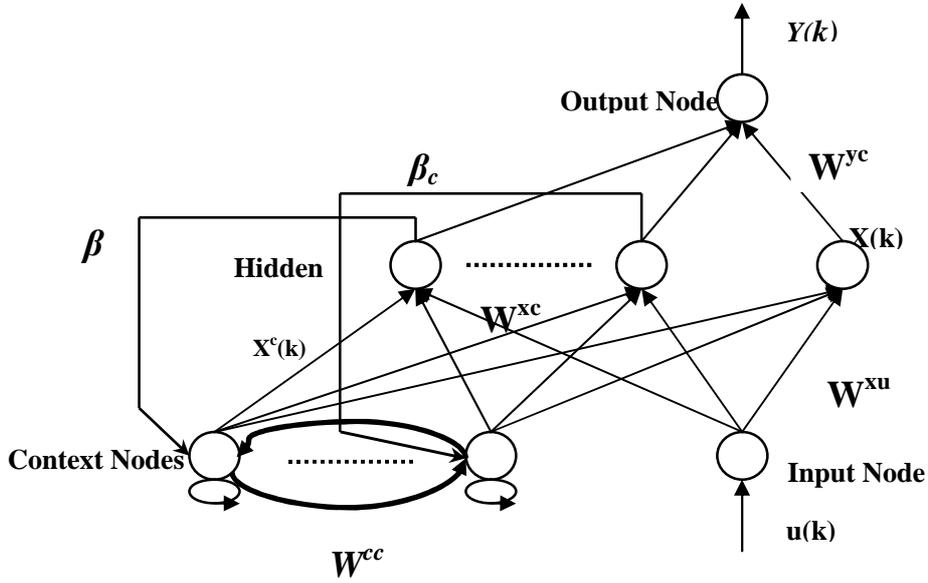
2 The Modified Structure of MRN

The MRN is modified to a new structure, which is called Interconnection Modified Recurrent Network "IMRN". The modification achieved by adding an adaptable weight to the context layer. IMRN has interconnection weights of context layer in addition to the adaptable weights from hidden to context layer. Fig. 2 shows the IMRN structure. The outputs of context layer nodes at time k have the form:

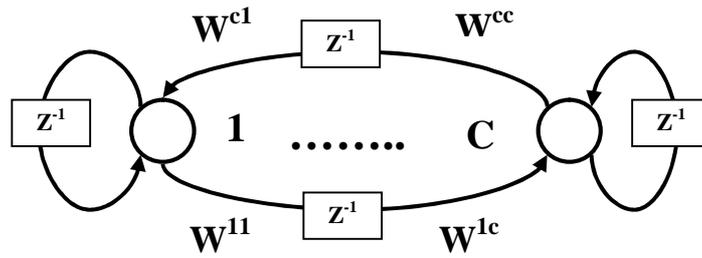
$$X_c^c = f \left[b_c X_n(k-1) - \sum_{c=1}^C \sum_{c=1}^C W_{cc}^{cc}(k) X_c^c(k-1) \right] \quad (1)$$

Where $(c=1, 2, \dots, C)$ is the number of context units. β is the adaptable weight from hidden layer to context layer. W^{cc} is the adaptable weight of the interconnection of context units. The

dimension of β is $(C \times 1)$ and the dimension of W^{cc} is square matrix $(C \times C)$. $X(k-1)$ and $X^c(k-1)$ is the past output of hidden and context layer respectively.



(a)



(b)

Fig. 2: Interconnection Modified Recurrent Network.

For the IMRN, the following equations can hold:

$$X_n(k) = f \left[\sum_{c=1}^C W_{nc}^{xc} X_c^c(k) + \sum_{p=1}^P W_{np}^{xu} U_p(k) \right] \quad (2)$$

$$X_c^c = f \left[b_c X_n(k-1) + \sum_{c=1}^C \sum_{c=1}^C W_{cc}^{cc}(k) X_c^c(k-1) \right] \quad (3)$$

$$Y_q(k) = g \left[\sum_{n=1}^C W_{qn}^{yn} X_n(k) \right] \quad (4)$$

Where W^{xc} , W^{xu} , W^{yx} and W^{cc} are the weights matrices, $f(\cdot)$ is a linear or nonlinear activation function and $g(\cdot)$ is a linear activation function, Y_q ($q=1, 2, 3, \dots, Q$) is the network output, ($n=1, 2, 3 \dots C$) the number of nodes in the hidden/context layers. ($U=1, 2, 3 \dots, P$) is the number of nodes in the input layer.

There are many learning algorithms available in the literature that can be used to train the IMRN (Sagiroglu, S., 1996). Among them the Backpropagation is the most commonly one used. This algorithm employs a gradient descent method to update the network weights that minimize the root mean square error between the network output and the desired one. Based on that, the updates of weights of the IMRN are given by following:

$$e(k) = Y_d(k) - Y_q(k) \quad (5)$$

$$\Delta W_{qn}^{yx} = h e(k) g'(Y_q(k)) X_n(k) \quad (6)$$

$$\Delta W_{nc}^{xc} = h e(k) g'(Y_q(k)) W_{qn}^{yx} f'(X_n(k)) \frac{\partial X_n(k)}{\partial W_{nc}^{xc}} \quad (7)$$

$$\Delta W_{np}^{xu} = h e(k) g'(Y_q(k)) W_{qn}^{yx} f'(X_n(k)) U_p(k) \quad (8)$$

$$\Delta W_{cc}^{cc} = h e(k) g'(Y_q(k)) W_{qn}^{yn} f'(X_n(k)) W_{nc}^{xc} f'(X_c^c(k)) X_c^c(k-1) \quad (9)$$

$$\Delta b_c = h e(k) g'(Y_q(k)) W_{qn}^{yx} f'(X_n(k)) W_{nc}^{xc} f'(X_c^c(k)) X_n(k-1) \quad (10)$$

$$g'(Y_q(k)) = 1 \quad (11)$$

Where, h is the learning rate and $Y_d(k)$ is the desired output.

3 System Identification and Simulation Results

The parallel identification scheme is the simplest among the system identification methods, which are based on neural networks. Its configuration is shown in Fig.3 (Kwok, D.P., et al, 1994). In this identification system, the IMRN works to gradually learn the model of the plant. The learning algorithm developed in the above section is used to train the network so that proper weights of trainable connections can be produced. The aim of learning is to minimize the RMS error with minimum epochs and computational time step by step with respect to

time. This identification system may use linear activation function when the plant is linear or nonlinear activation function when the plant is nonlinear.

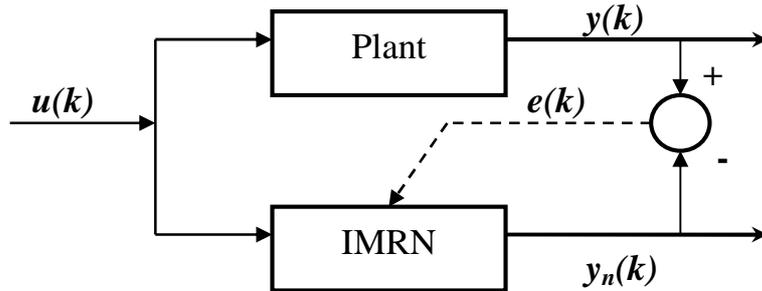


Fig.3: System identification using the IMRN.

The performance of IMRN for system Identification will be examined in the following examples. Two systems were considered; one linear while the other was a nonlinear system. It was assumed that the two plants are bounded input bounded output stable. The response of the IMRN is then compared with the responses of the MRN and MENN. All the three structures have single input node, six nodes in the hidden/context layer and single output node. The learning rate was chosen by trail and error.

3.1 Example -1: Identification of Second-Order linear Plant

The plant is described by the equation:

$$y(k) = 1.75y(k-1) - 0.818y(k-2) + 0.01169u(k-1) + 0.0109u(k-2)$$

During the training phase, random signal was presented to the input. Meanwhile, during the test phase, different signal was used. Tab.1, illustrates the RMS error, number of training cycles and the computational time for the three types of networks. It is obvious; the IMRN gives the finest performance. Simulation results are shown in Fig. 4 to Fig. 7. It is clear, Fig. 4, that the IMRN reached the required error goal faster than the other two networks. The responses of the plant and the three networks to the same input signal are plotted in Fig. 5 to Fig. 7. As expected, the IMRN gives better response than the other two networks.

Table 1: Comparison between the performances of the IMRN, MRN and MENN.

Network	Parameters		
	RMS error	No. of Cycles	Computational Time (Sec)
IMRN	0.0002	10	16.64
MRN	0.0011	70	104.74
MENN	0.025	120	81.35

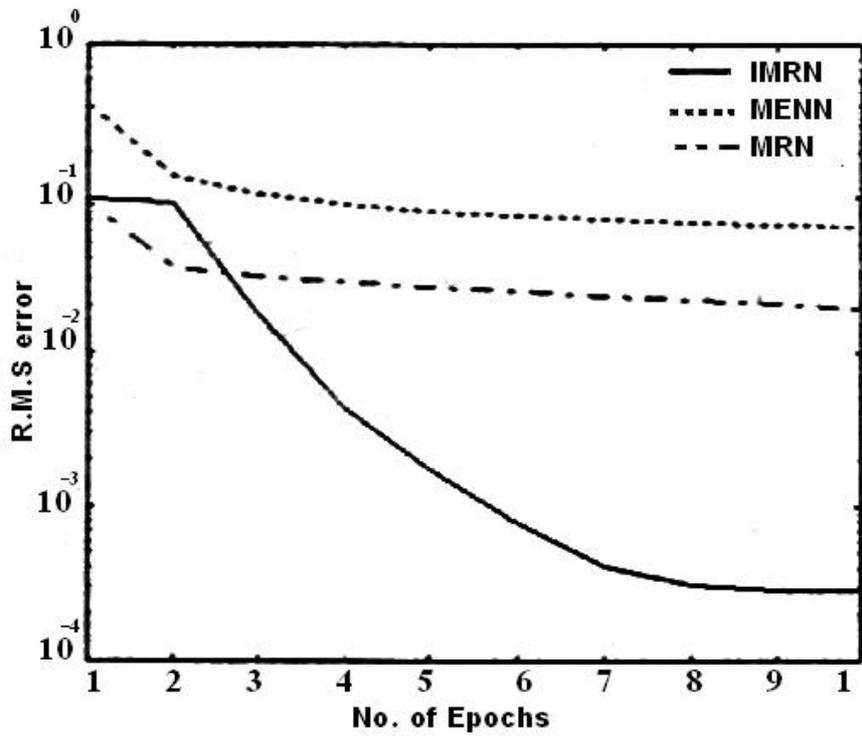


Fig. 4: Training RMS error of example 1.

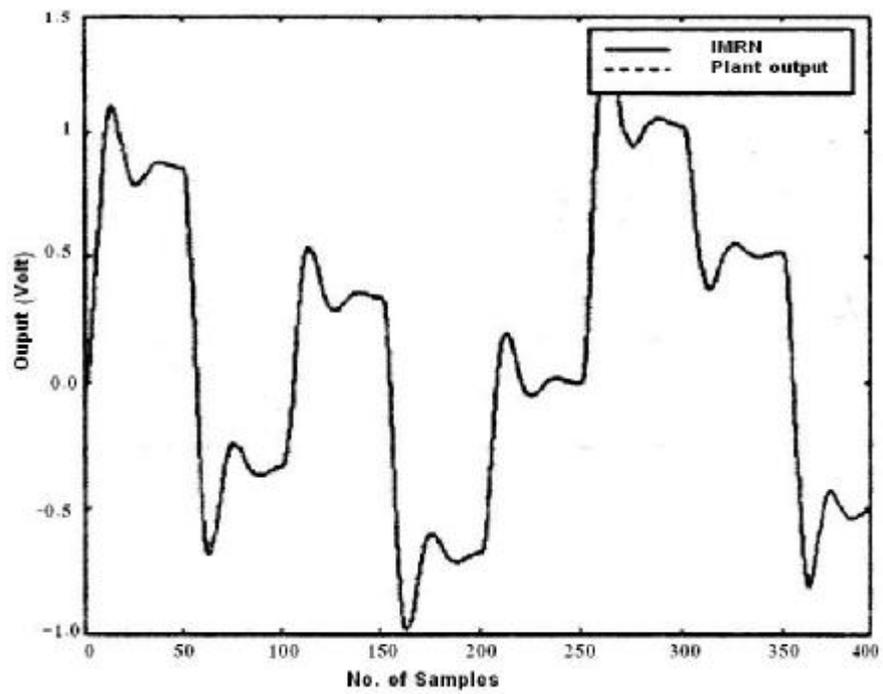


Fig. 5: Response of the IMRN and the plant.

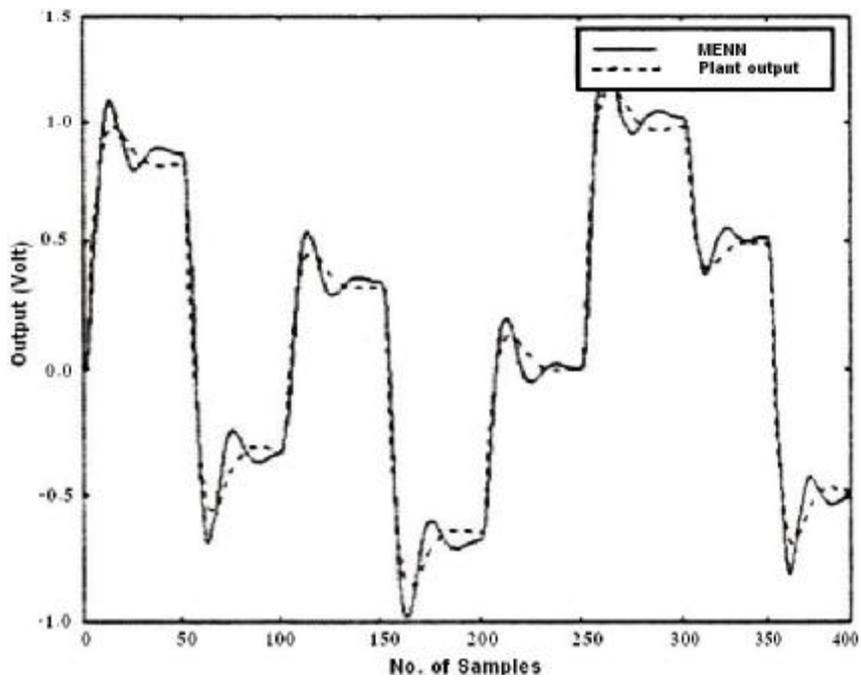


Fig. 6: Response of the MENN and the plant.

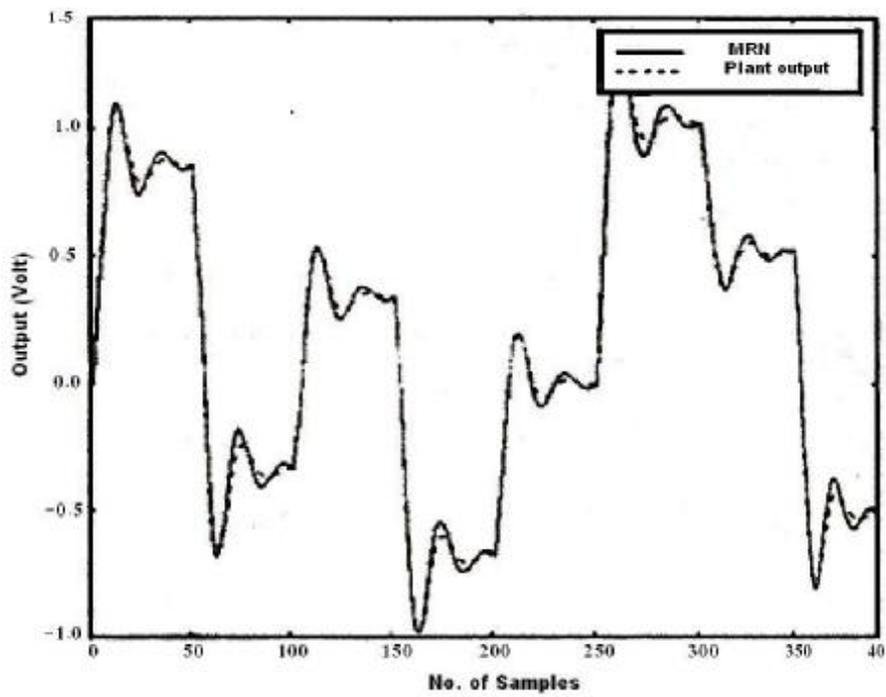


Fig. 7: Response of the MRN and the plant.

3.2 Example -2: Identification of Second-Order nonlinear Plant

The plant is described by the equation:

$$y(k) = 0.35 \left[\frac{y(k-2)y(k-1)^2 + 2.5}{1 + y(k-1)^2 + y(k-2)^2} + u(k-1) \right]$$

Once again, during the training phase, random signal was presented to the input. Meanwhile, during the test phase, different signal was used. Tab. 2, illustrates the RMS error, number of training cycles and the computational time for the three types of networks. It is noticeable that the IMRN gives the best performance. Simulation results are shown in Fig.8 to Fig.11. It is clear, Fig.8, that the IMRN reached the required error target faster than the other two networks. The responses of the plant and the three networks to the same input signal are plotted in Fig.9 to Fig.11. As expected, the IMRN gives better response than the other two networks.

Table 2: Comparison between the performances of the IMRN, MRN and MENN.

Network	Parameters		
	RMS error	No. of Cycles	Computational Time (Sec)
IMRN	0.007	300	818.12
MRN	0.0073	400	1179.1
MENN	0.0151	400	931.173

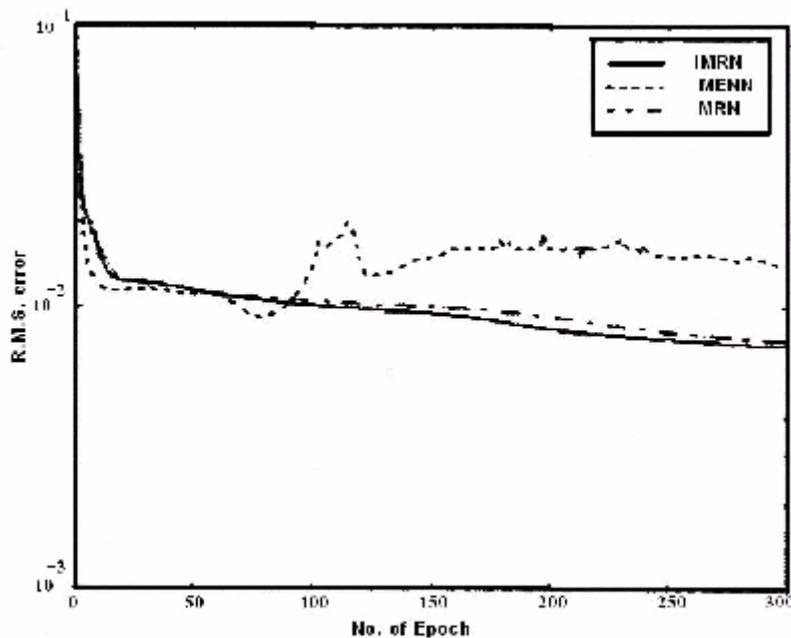


Fig. 8: Training RMS error of example 2.

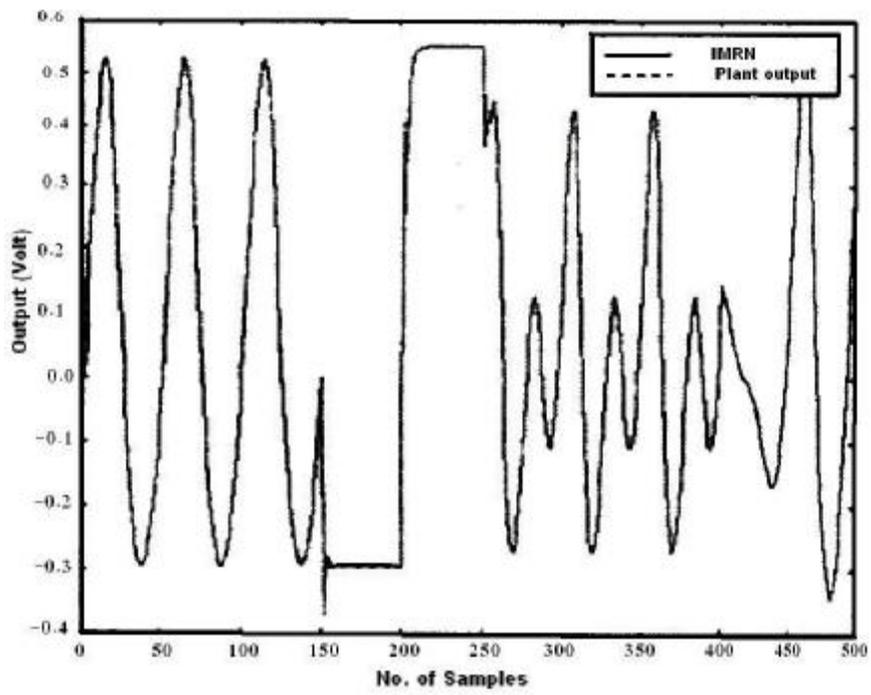


Fig. 9: Response of the IMRN and the plant

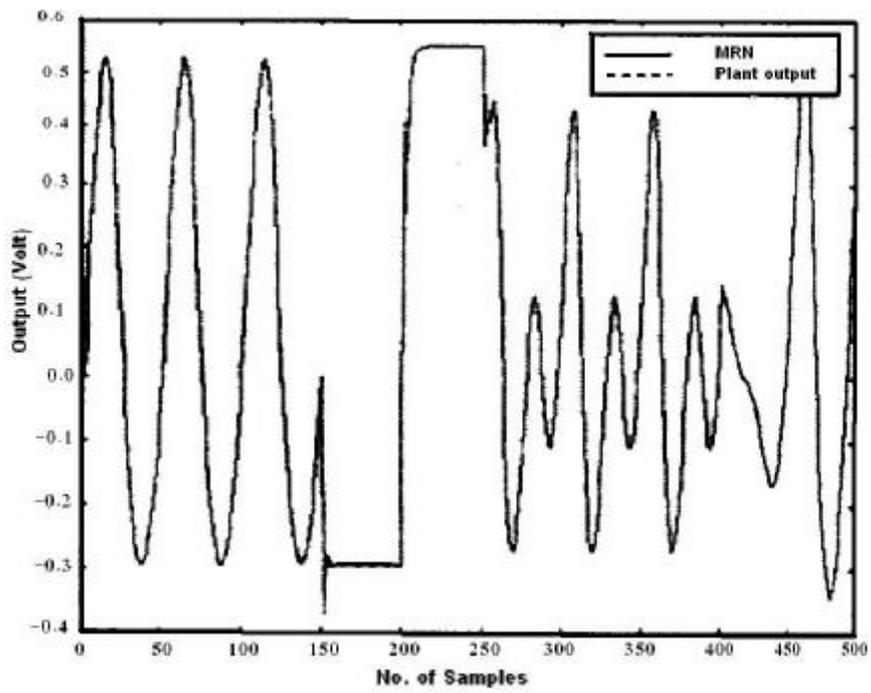


Fig. 10: Response of the MRN and the plant.

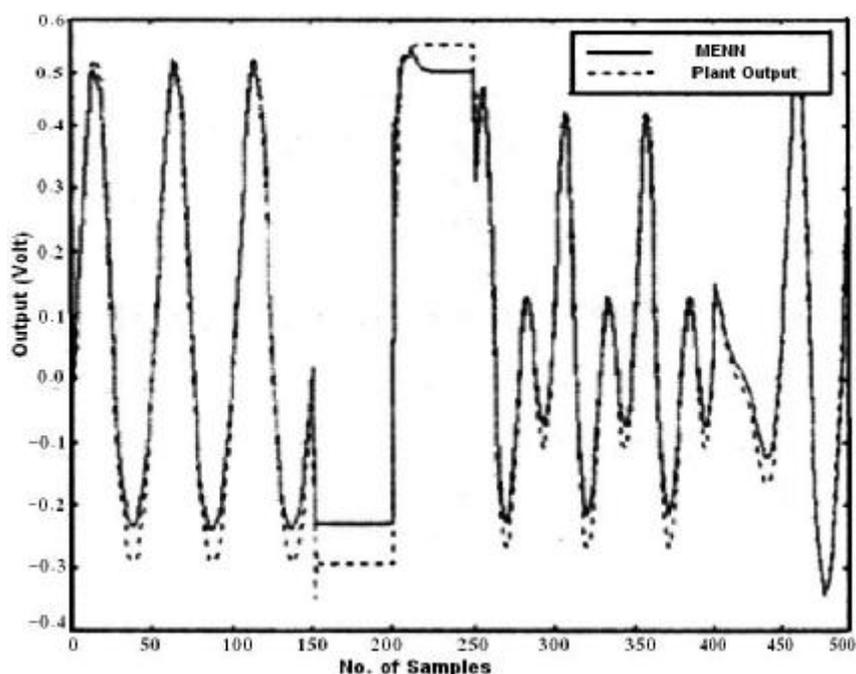


Fig. 11: Response of the MENN and the plant.

4 Conclusions

A modified structure of PRNNs was proposed in this paper. The standard BP training algorithm was used to train the IMRN. The proposed structure with its training algorithm overcomes most of the disadvantages of the PRNNs that have fixed self-connection weights and fixed feedback-connection weights. The performance of the IMRN was simulated, tested and compared with that of the MRN and MENN. The three neural structures were used in the identification process of linear and nonlinear plants. Simulation results proved that the IMRN reach the error goal much faster than the MRN and the MENN in terms of training cycles and computational time.

References

- AL-Faysale, M.S.M., 2004, A Training Algorithm for Partial Recurrent Network, *Proceeding of the Middle East Multiconference on Simulation and Modeling (MESM2004) Conference*, (Philadelphia University, Amman, Jordan), 194-198, 14-16 September, 2004.
- AL-Faysale, M.S.M., 2005, Partial Recurrent Network for System Identification and Control, *Proceeding of the Third IEEE International Conference on Systems, Signals and Devices (SSD'05)*, Sousse, Tunisia, 22-24 March 2005.
- Kwok, D.P., Wang , P. and Zhou, K., 1994, Process Identification Using a Modified Elman Neural Net., *In International Symposium on Neural Networks*, Hong Kong, 499-501, 13-16 April 1994.

- Lin, T., Hene, B.G. and Giles, C.L., 1998, How Embedded Memory in Recurrent Neural Network Architectures Helps Learning Long-Time Temporal Dependencies, *IEEE Trans. NN*, Vol.11, No.5, 861-868, 1998.
- Peddersen, M.W., 1997, Training Recurrent Networks, *In Neural Networks for Signal Processing, Proceeding of the IEEE Signal Processing Society Workshop*, 335-364, 1997.
- Pham, D.T. and X. Liu, 1997, *Neural Networks for Identification, Prediction and Control*, Springer-Verlag, New York, 1997.
- Sagiroglu, S., 1996, Linear System Identification Using Multilayer Perceptron," *Proceedings of ISCIS XI*, 2, Turkey, 517-527, November 1996.
- Santosand, J. and Duro, R.J., 1998, Learning to Reason in Time Using Discrete Time Back-propagation, *Proceeding of the International Conference on Neural Networks and Brain*. 1998.
- Sastry, B.S., Santharan, G. and Unnikrishnan, K.P., 1994, Memory Neural Networks for Identification and Control of Dynamical Systems, *IEEE Trans. NN*, Vol.5, No.2, 306-319, 1994.

Computer Aided Modelling of a Mechatronic System

M. Ebrahimi, A. A. Abdul-Ameer and R. Whalley
School of Engineering, Design and Technology
University of Bradford
UK

ABSTRACT *Mathematical models are frequently used in the design, analysis and development of engineering systems. This paper looks at the ways that commercial Computer-Aided Control System Design (CACSD) software packages can be used to automate the design, calculation and modelling process. A typical servo-mechanism system is modelled in block diagram form. The investigation of the model (design) using CACSD software shows some of the capabilities on offer to the designer.*

Key words: *engineering, system, analysis, modelling and simulation*

1. INTRODUCTION

During the past two decades, the falling price of the computer peripherals and the development of commercial Computer Aided Engineering software packages have advanced mathematical modelling and simulation methods. With abundant computer power, and inexpensive CAE software, design, modelling and system improvement are accessible to most small to medium manufacturers. The motivation for this use can be easily justified in cost efficiency. This stems from the fact that once committed to modelling and simulation there are savings in product development and improvements, the merits of which are well known to the defence industries.

Also in the design and construction of machine tool feed drive systems for example, it is desirable to predict the effects of the machine and the non-linear parameters commonly encountered such as, backlash and friction on the servo system performance (Younkin, G. W 1991). The cost for redesigning and testing can be avoided, if the effects of these non-linear parameters can be minimized or eliminated prior to manufacturing. Currently there are many commercial software packages such as MATLAB, CTRL and MATRIX_X that are used in different capacities. Most of these packages can handle systems with elements which are non-linear, continuous, discrete and multi-rate. Using a set of element blocks available in aforementioned CADSD packages, a non-linear model can be constructed for simulation purposes. The element blocks available cover an entire range of the control elements necessary to describe a system model. The data for manipulation can be obtained from the model or imported from other programs. This enables the actual signal from the System Under Test (SUT) to be obtained by data acquisition system for analysis purposes.

Time domain representations of control systems started in the early 1960's. This resulted in the development of state-space techniques which gradually became the principal tool for solving multivariable systems problems. In classical linear control the physical system can also be modelled as single input single output (SISO), linear, continuous and time invariant model. To analyse this type of systems time and frequency domain methods have been developed. These linear models can be further expanded (if required) by adding system non-linearities. To analyse non-linear models, techniques such as describing functions and phase plane methods have been developed. SISO are usually presented by transformed differential equations, as a transfer functions. With increasing degrees complexity a need for multivariable system methods were required. However if a SISO

system is required to portray more than one input and output then as a result a multi-input multi-output (MIMO) system has to be defined. This resulted in development of both state-space transfer function methods to describe MIMO systems.

The aim of this work is to use CACSD software to develop a mathematical model of a machine tool servo-feed drive system that can be easily applicable for various fields of design and research.

2. MODELLING TECHNIQUES

The modelling process can be divided to physically and empirically synthesise the system, as shown in Fig. 1. The classical procedure for the design of such a system is usually the modification of the existing design which is based on physical laws. The designer then calculates the new system specifications based on the modified parameters. The numerical value of the elements of the plant can be obtained from the literature; or derived from geometrical and material constants. This information is converted into a set of first order, ordinary -differential equations or state equations.

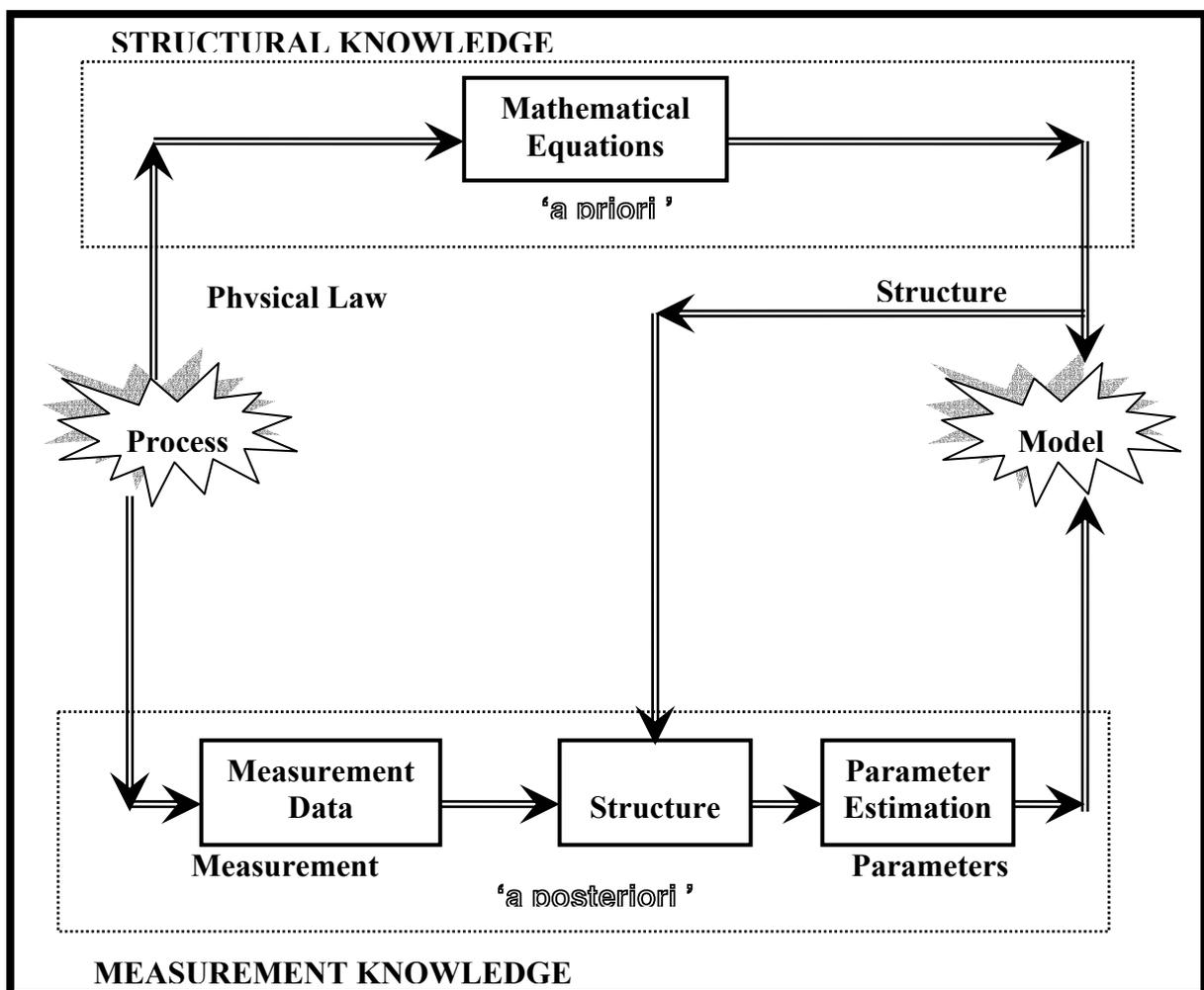


Fig. 1, Illustration of a modelling procedure

The block diagram modelling can be envisaged as a medium that conceptualises the mathematical equations in a pictorial way by using the signal between different elements of the system. The advantages of using the block diagram model can be classified in to 1) Its conformity with most

engineering disciplines 2) ease of programming 3) availability of block diagram manipulation within most of the control packages.

Due to the factors which need consideration, there are many ways in which modelling can be classified and formalised, (Eykhoff, P. 1989 and 1974). Essentially there is one approach in finding a model, which is used in this work. This approach is the “Lumped Parameter Modelling (LPM) technique. The subsystem is broken down into lumped pointwise elements for which a mathematical description is then established.

2.1 Lumped Parameter Modelling (LPM)

In lumped parameter modelling (LPM) an idealised equivalent, physical model of the system, is considered, which each element of the model has a single property. The advantage of considering such an idealised or 'lumped parameter' system is that each element has one independent variable - time, so that the system can then be described using a differential equation model.

In a time-invariant, lumped parameter linear system any response or output signal $Y(t)$, due to any external excitation or input signal, is linked to the corresponding excitation by a differential equation with constant coefficient, of the form:

$$a_0 \frac{d^n U(t)}{dt^n} + a_1 \frac{d^{n-1} U(t)}{dt^{n-1}} + \dots + a_n U(t) = b_0 \frac{d^m Y(t)}{dt^m} + b_1 \frac{d^{m-1} Y(t)}{dt^{m-1}} + \dots + b_m Y(t) \quad (1)$$

where U and Y are the input and output, and the terms $a_0, a_1, a_n, b_0, b_1, b_2, \dots$ are constant parameters unique to a given system. If the parameters of the generated differential equation are not constant then the equation is time varying. It is possible to model the most physical systems by LPM techniques. However there are situations where LPM leads to a high order description which requires model reduction methods. Alternatively identification technique can be used, if the system exerts as an alternative approach to LPM.

2.2 System Identification (SI)

System Identification is the process of constructing a mathematical model of a dynamical system from observation and 'a priori' knowledge. The purpose of SI is to determine the dynamics of a system by means of experiment. The following definition is given by (Zadeh, L. A. 1962), "Identification is the determination, on the basis of the input and output, of a system (model) within a specified class of systems (=models), to which the system (=process) under test is equivalent".

2.3 Transfer Function Models (TFM)

The above modelling exercises results in a number of sub- systems or elements, each of which has a differential equation representation. In addition to the differential equation, the transfer function is another way of mathematically modelling a system. The modelling in this case is concerned more with forced response than with transients due to initial conditions. If the system is stable, the influence of the initial condition on the output becomes negligible as time progresses. It is therefore common practice to assume all the initial condition are zero, in which case the transfer function of a linear system $G(s)$ can be written as the ratio of an output $Y(s)$ developed by an input $U(s)$; Hence

$$Y(s) = U(s) * G(s) \quad (2)$$

The application of block reduction techniques, or 'block diagram algebra' condenses the Laplace transform system equations into a form suitable for modelling exercises. The z -transform plays a similar role in discrete time systems, to that of by the Laplace transform in continuous system transfer function analysis. The difference equation of a system may be converted into transfer function by the use of the z -transform in a similar way as a differential equation into s -transform by applying the Laplace transform. The transfer function $G(z)$ for a discrete-time system is defined as that factor in the equation for the transform of the output $Y(z)$ that multiplies the transform of the input $U(z)$. If all the term due to initial conditions are zero, then the response to an input $U(z)$ in z -domain is given by:

$$Y(z) = G(z) * U(z) \quad (3)$$

2.4 Block Diagram Algebra (BDA)

Block diagrams have been used to represent Laplace transform, z -transform equations and describing functions for machine tool elements. The use of block diagram is a convenient method of pictorially grouping these elements such that the overall model of a subsystem and finally the machine tool itself can be represented in a mathematical meaningful way.

3. SERVO SYSTEM OVERALL MODEL

To perceive the most suitable simulation model of a machine tool, servo-feed drive system, it is necessary to know the reason for, and the application of, the model. The aim of this work is to present a simulation model which can be used to predict different operating conditions for machine tools. The simulation model allows the machine designer/researcher to observe the effect of changing the drive parameters such as, backlash, viscous friction, load inertia ... etc. on the feed drive system performance without having the physical drive system available. Therefore, the objective of the simulation model is to provide insight in to the operation of the milling machine tool defining the cause-and-effect relationship between variables. The important relationship is that between the manipulatable input and measurable output. This encompasses the interrelationship of the movement of the workpiece during cutting and the forces and torques acting on the x axis feed drive system. The particular objective of this work is then to present a mathematical model which explores the dynamics of the x axis feed drive.

The x -axis assembly comprises an A.C, permanent magnet, synchronous motor which provides the torque, applied to the lead-screw, at the drive end. In large machines the lead-screw may be several metres long with a relatively small cross sectional area where the effective diameter is measured at the root, of the lead screw thread.

Ball and thrust bearings support the lead –screw, attached to which is the machine saddle and workpiece. A ball-screw assembly is normally used to transfer the motion from the lead-screw rotation to the machining saddle.

This assembly is shown in Fig. 2, where the principle components are labelled. The force – torque directions are also shown in this diagram enabling the various combined twisting moments and torsional forces to be identified.

3.1 The Drive Servo-Motor

The drive motor is directly coupled to the lead-screw, as shown in Fig. 2. This motor is a permanent magnet synchronous machine (PMSM), with an integrated encoder.

PMSM have a rotor with a set of three phase windings similar to those in three phase induction motors, see for example (Pillay, P and Krishnan, R 1989). The three phase voltage supply, applied to the windings of a PMSM, produces constant amplitude, rotating m.m.f in the air gap. Rotor

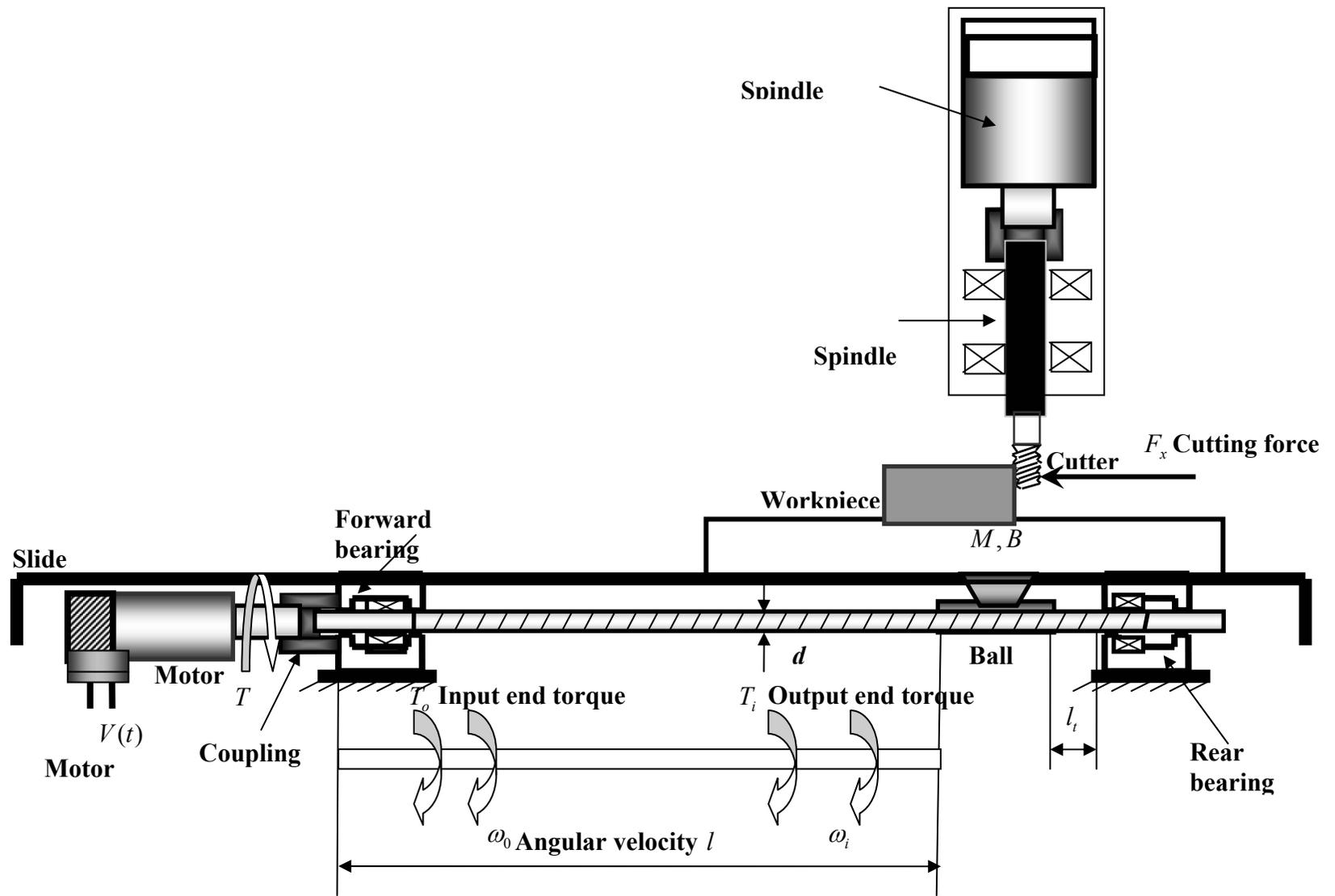


Fig. 2, Machine tool servo-system, x-axis drive system

angular position feedback is used to synchronise the rotor position, relative to the stator winding, current frequency. The shaped permanent magnets forming the rotor provide an approximate sinusoidal flux which interacts with the stator flux to provide the electromagnetic turning moment required.

For analytical and modelling purpose PMSM are usually transformed from three to two phase devices. Essentially, this enables models to be constructed using either a two-axis, synchronous rotating (d, q) reference frame or a stationary (α, β) frame may be employed, see for example (Husain, I. 2003).

The relative, angular location of these axis, are shown in Fig. 3. The coordinate transformation enabling transfers from each three, to two phase system, are also indicated in this figure.

Reference axis d - q rotate with the rotor whereas the α - β coordinates remains stationary with the α - β axis sharing a common, x -axis basis. The β axis lags the α axis by 90° and the 3 phase a - b - c vectors lag each other by 120° . The d vector is in the direction of the direct, and the q vector is the direction of the quadrature flux, in the equivalent d - q coordinate system.

In the modelling process, for an equivalent two phase machine, eddy current and hysteresis losses are neglected. The equations representing the machine's performance in the d - q coordinate system are, as shown by (Sebastain, T and Slemon, G. R. 1989):

$$\frac{di_d(t)}{dt} = \frac{1}{L_d} V_d(t) - \frac{R_s}{L_d} i_d(t) + \frac{L_q}{L_d} p \omega_0(t) i_q(t) \quad (4)$$

$$\frac{di_q(t)}{dt} = \frac{1}{L_q} V_q(t) - \frac{R}{L_q} i_q(t) p \omega_0(t) i_d(t) - \frac{\lambda p}{L_q} \omega_0(t) \quad (5)$$

$$T_e(t) = 1.5 p \left(\lambda i_q(t) + (L_d - L_q) i_d(t) i_q(t) \right) \quad (6)$$

$$T_e(t) - T_b(t) = T_0(t) \quad (7)$$

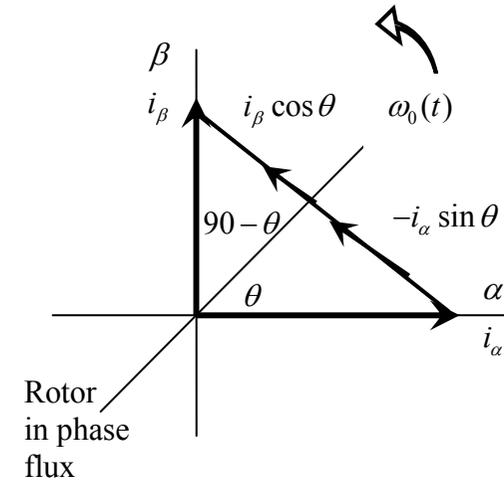
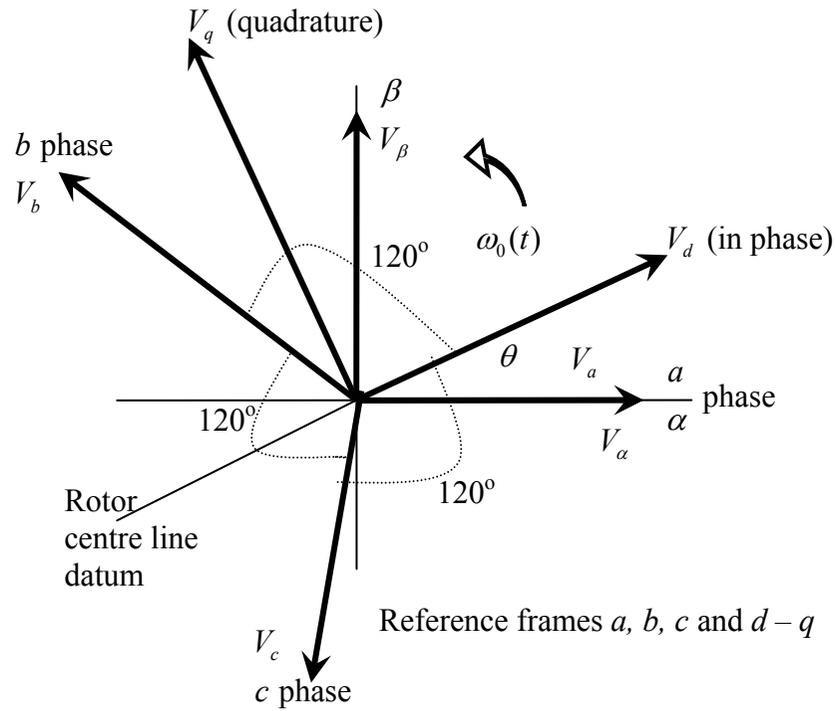
$$\frac{d\theta(t)}{dt} = \frac{p}{2} \omega_0(t) \quad (8)$$

Following Laplace transformation with zero initial conditions, Fig. 4, can be constructed, showing the block diagram representation of equations 4 to 8.

The PMSM is fed from a power, electronic inverter, for efficient operation of the system. Fig. 5 shows the drive motor system in block diagram form, including the inverter and inner loop speed controller.

3.2 Axis Drive System Dynamics

The free body diagram of the axis drive system is represented by Fig. 6, where B_1 is the viscous damping coefficient of the motor, B_2 is the viscous damping coefficient of the ball screw bearings and ball nut. J_1 is the moment of inertia of the motor, and J_2 is the moment of inertia of the lead screw. T_F is the coulomb friction of the motor, ball screw, bearings, and the ball nut. θ' represents the motor speed and consequently the lead screw. θ represents the angular position of the motor and lead screw. While, θ_0 is the shaft position at the ball nut, the difference between the ball screw angular position θ and θ_0 at the ball nut is referred to as the angular deflection of the lead screw. The rotational movement of the ball screw leads to a linear displacement of the nut and consequently the table, X_0 . The mass M is given by the mass of the table and the spindle drive system. T_l is the torque generated in the motor armature. T_0 is the drive torque applied by the ball screw nut to the table. T_D provides the force F_D , acting on the table, which is required to provide



$$\begin{bmatrix} V_d \\ V_q \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \theta & \cos(\theta - 2\pi) & \cos(\theta + \frac{2\pi}{3}) \\ \sin \theta & \sin(\theta - \frac{2\pi}{3}) & \sin(\theta + \frac{2\pi}{3}) \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

Park transformation from $a-b-c$ frame to $d-q$ frame

$$\begin{bmatrix} V_\alpha \\ V_\beta \\ 0 \end{bmatrix} = \frac{2}{3} \begin{bmatrix} 1 & -0.5 & -0.5 \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

Park transformation from $\alpha-\beta$ frame to $a-b-c$ frame

Fig. 3 Rotating 3 phase voltage vectors, d, q reference frame and α, β fixed reference frame

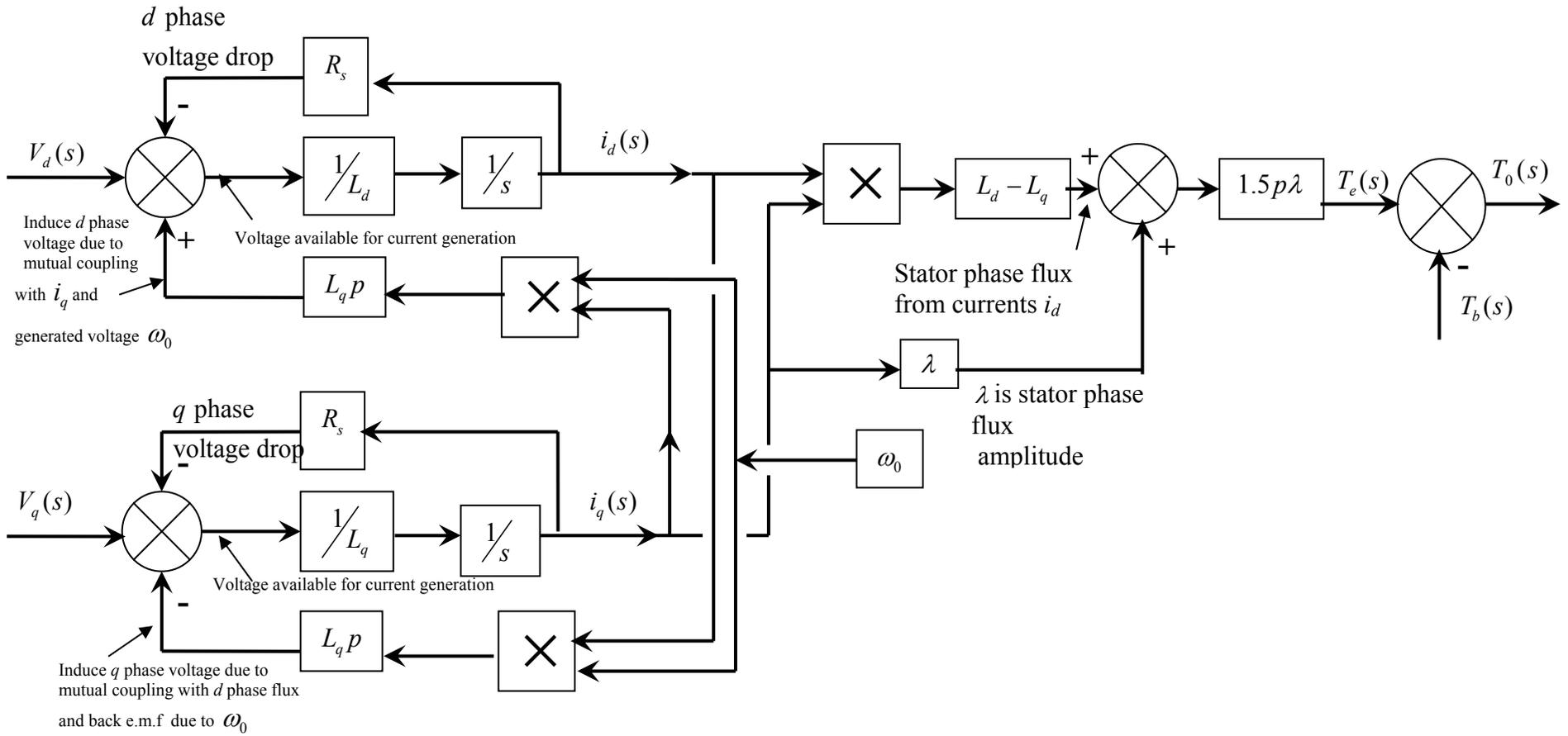


Fig. 4 Permanent magnet synchronous motor block representation with d, q reference frame)

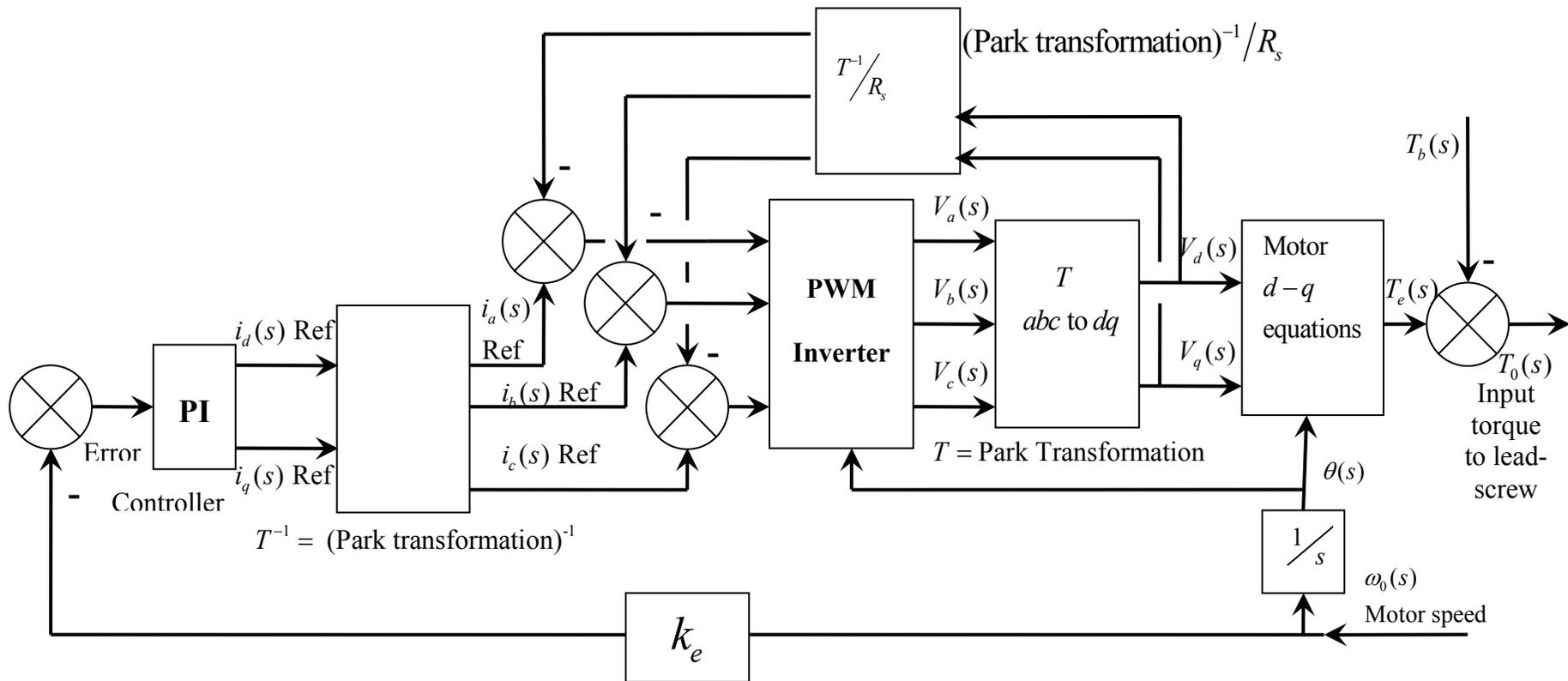


Fig. 5 Drive motor, block representation

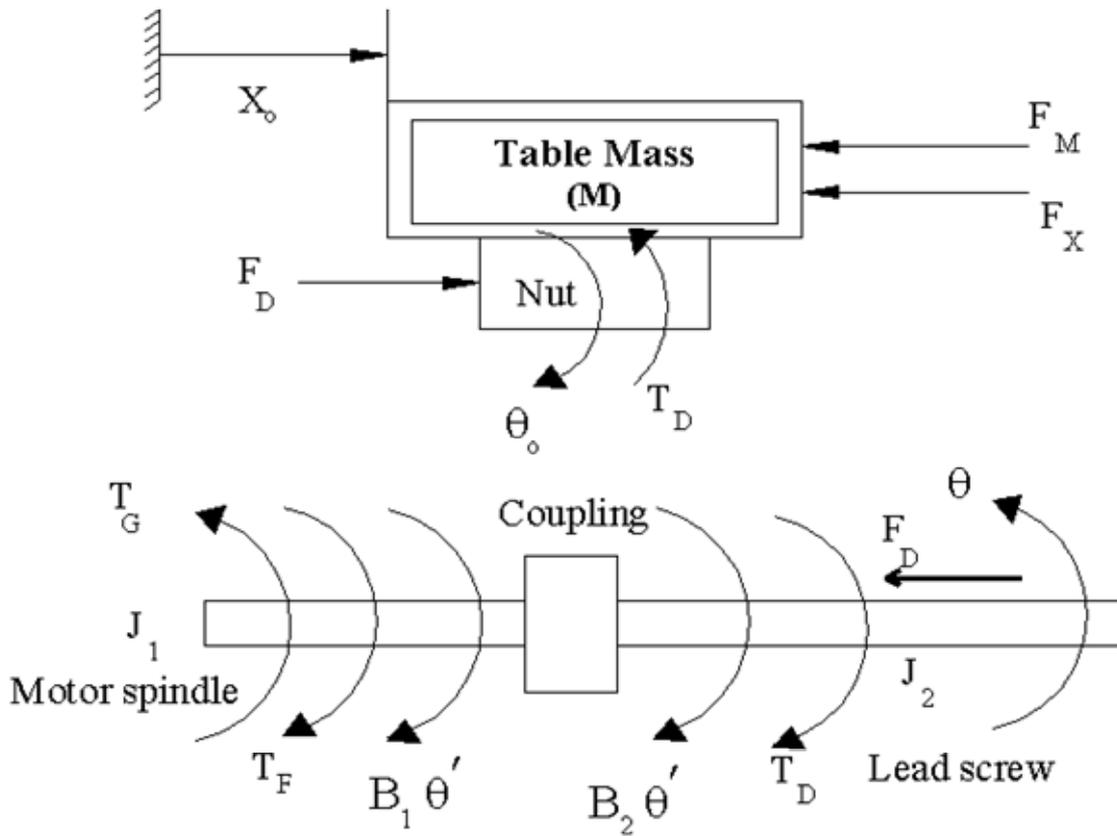


Figure 6, The free body diagram of the X-axis drive system

the force F_M to accelerate the table and overcome friction, and the axial cutting force component F_X . $B_1'\theta$ and $B_2'\theta$ are the resistance due to viscous friction of the motor and lead screw, respectively.

3.2.1 Lead Screw and Nut Mechanism

The lead screw and nut are designed to transfer the rotary motion (the motion of the lead screw) into linear motion (the motion of the nut). The required torque on the lead screw to overcome the axial force F_D can be:

$$T_D = \frac{p}{\eta 2\pi} F_D \quad (9)$$

where, p is the pitch or lead of the ball screw (m), and η is the lead screw efficiency, which is equal to 100% as it is taken into considerations in B_R and T_F .

$$T_D = K_1 F_D \quad (10)$$

where, K_1 is the force to torque constant (m).

The rotation of the ball screw accounts for a linear movement of the nut by X . This longitudinal motion is proportional to the angular motion of the ball screw at the nut θ_0 .

$$X = K_2 \theta_0 \quad (9)$$

3.2.2 Stiffness in Axis Drive System

Stiffness plays an important role in the accuracy of the machined tools. The overall stiffness accounts for the combination of the elastic parts in the mechanical chain of the axis drive system. To obtain the actual linear displacement of the table, it is essential to take account of the deflection due to the angular and linear stiffness (Ebrahimi, M. and Whalley, R. 2000).

3.2.3 Backlash in the Axis Drive System

The hysteresis in machine tools is represented by backlash which means that the driven member, the ball-screw nut, remains stationary when not in contact with the input member, ball-screw. The backlash in the axis drive system is represented by a frictional-controlled backlash element, where, for small movements of the input (input < backlash), about the central datum, there will be no output movements (Richards, R. J.1979). It is assumed that the backlash in the ball-screw is represented as a backlash in the axis drive system. The backlash in machine tools affects motion accuracy. There is a delay in the response of the table following the rotation of the axis drive motor (Kao, J. Y. et al. 1996).

3.3. Table and Guide-Ways

The movement of the table over the guide-ways is subjected to viscous and coulomb friction. To move the table, the longitudinal drive force F_D is equal to the sum of the axial cutting force (feed force F_X) and the force F_M required accelerating the table and overcoming the friction force. Fig. 7 represents the free body diagram of the table and saddle, considered as a lumped mass M .

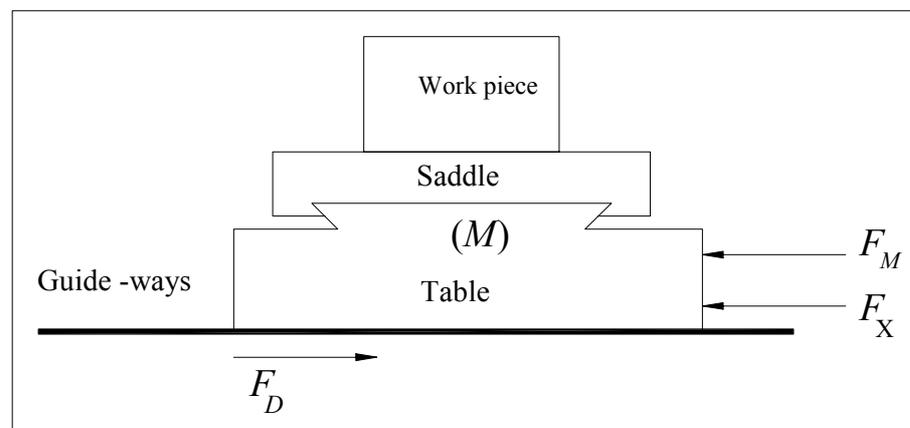


Fig. 7, Free body diagram of the table and guide-ways

3.4. Overall Axis Drive System Model Block Diagram

The overall axis drive model, block diagram, can be obtained from connecting all the sub-systems of all elements in the system. These elements contain the mechanical properties of the axis drive motor, ball-screw, the ball-screw nut mechanism, backlash element and the table. This overall axis drive model is presented in Fig. 8.

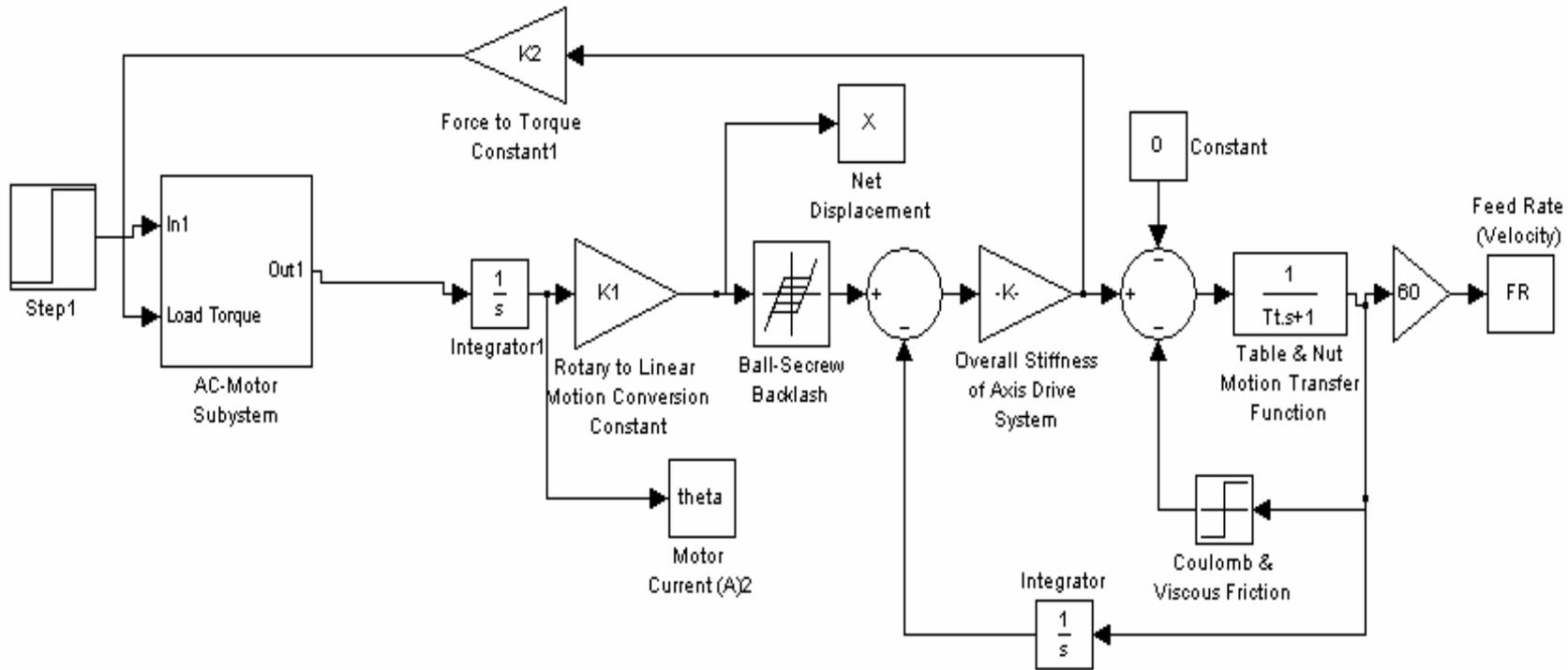


Fig. 8, The axis drive system overall simulation model

4. Simulation Results

Simulation is the technique whereby a model of a real system can be constructed, in order to study its behaviour, without disturbing the environment of the real system. Simulation is also the process whereby important aspects of the behaviour of a system, in real time, can be studied. Computer simulation means the running of a special program on a computer, which generates time responses of the model imitating the behaviour of the process under consideration (Law, A. M. 1991). Simulation can be used to determine the performance of machine tool subsystems by changing the model parameters. Also, the simulation programs enable researchers to observe the performance of the system without having to change the actual parameters of the machine tool. Figures 9-A and 9-B present the comparison between the experimental and the simulation results for a CNC milling machine.

Discussion & Conclusion

The procedures, enabling the construction of a lumped model for the x-axis dynamics for a machine tool system, were presented in this paper. Only the lead-screw, for the traverse mechanism, was described in distributed parameter form, in view of its high length-diameter, slenderness ratio.

The remaining elements including the saddle, ball-nut, motor-drive etc., were all considered to be relatively concentrated, in terms of their mass-inertia, damping and stiffness properties. Therefore, these components were modelled using conventional lumped parameter methods.

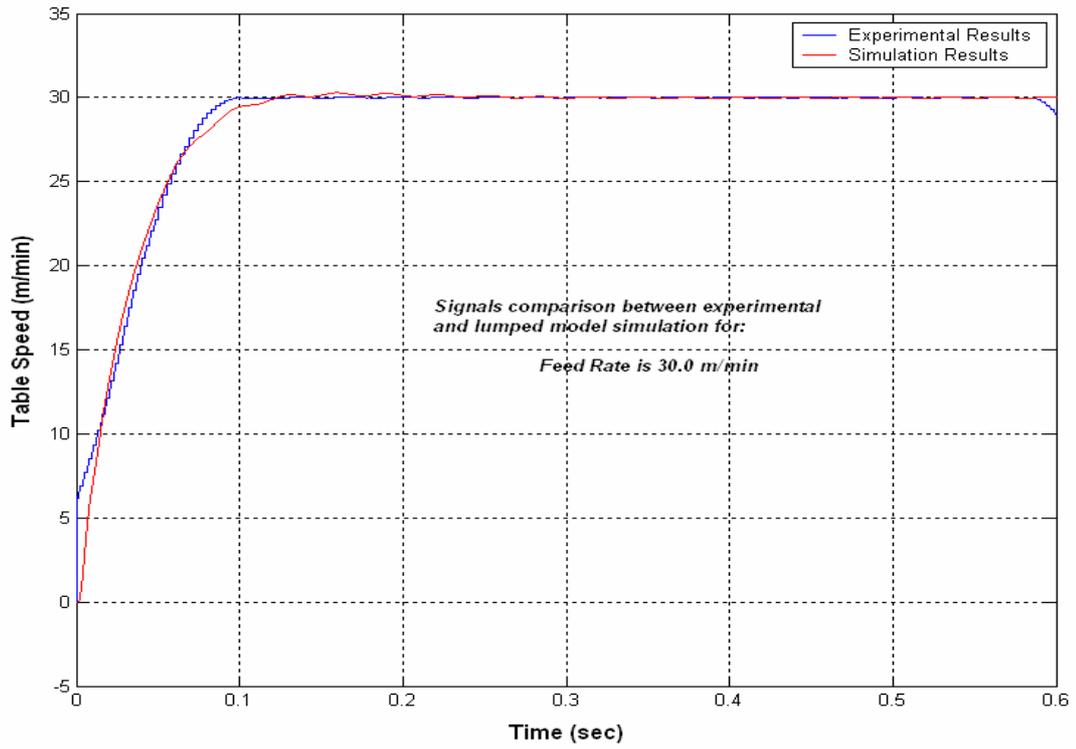
To illustrate the formulation techniques involved the machine tool elements were shown as block diagram, sub-assemblies. These realisations were then arranged to represent the overall system model.

This model generates the transient effects arising from spatial dispersion owing to the lead-screw dimensions. Both the torsional response characteristics and the loading deflection, arising from lead-screw “wind-up” were included in the modelling procedure.

These effects are amplified, when the system model was excited by input changes and cutting force disturbances. Significantly, the results obtained demonstrated that the model replicated the broadband dynamics of the system, comprising the low frequency response, steady state dynamics and the internally generated machine tool chatter.

The table speed variations caused by changes, in the saddle position and/or feed rate, could also be easily investigated, if desired. Moreover, although not specifically pursued, the vibrational disturbances arising from alternative cutting tools as a results of transient deflections, could be simply incorporated following an analysis of the transverse vibrations of these units. When using CACSD great attention must be paid to numerical stability, otherwise the results will be worthless.

A. Table Speed



B. Motor Speed

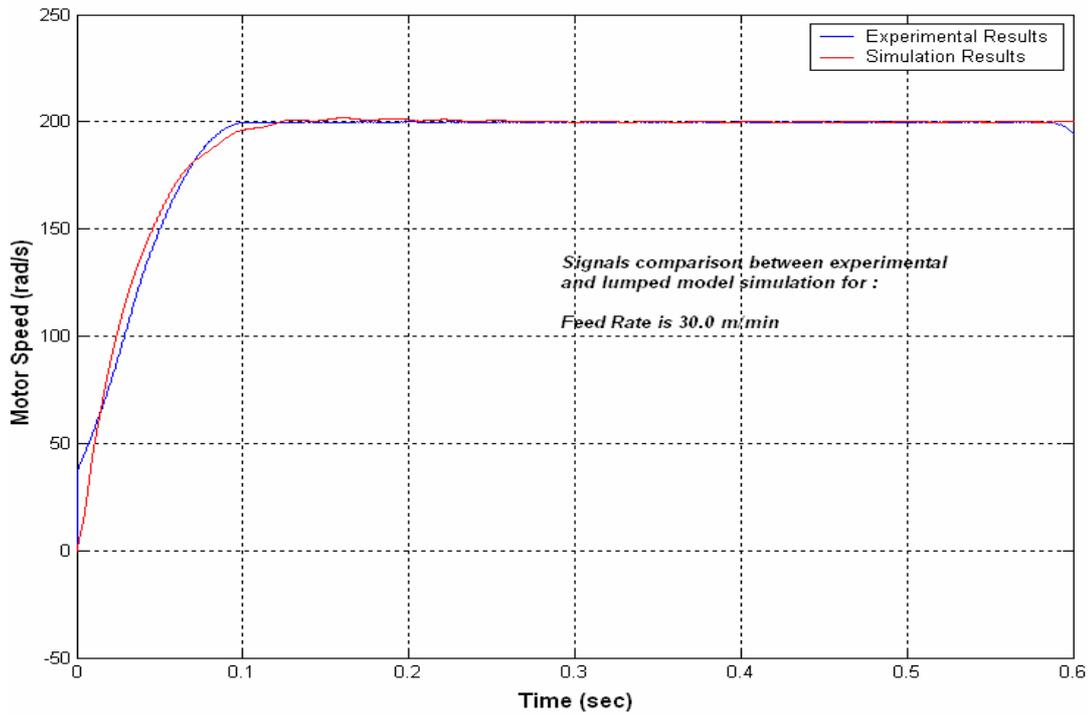


Fig. 9, Signal comparison between experimental and simulation results for CNC milling machine axis drive system

References:

1. Ebrahimi, M. and Whalley, R., "Analysis, Modelling and Simulation of Stiffness in Machine Tool Drives", *Computers and Industrial Engineering*, Vol. 38, 2000, pp. 93 – 105.
2. Eykhoff, P., "A Bird's-Eye View on Parameter and System Identification", *Soviet. J. Autom. Inf. Sci.*, March 1989, Vol. 20, No. 6, pp 15-38.
3. Eykhoff, P., "System Identification, Parameter and State Estimation", Wiley, London, 1974.
4. Husain I., "Electric and Hybrid Vehicles, Design Fundamentals", 2003, CRC Press LLC, Boca Raton, Florida, USA.
5. Kao, J.Y. *et al.*, "A Study of Backlash on the Motion Accuracy of CNC Lathes", *International Journal of Machine Tools and Manufacture*, Vol. 36, no. 5, 1996, pp. 539-550.
6. Law, A.M., "Simulation Modelling and Analysis", 2nd Edition, McGraw-Hill, Inc., 1991.
7. Pillay, P. and Krishnan, R., "Modelling, Simulation, and Analysis of Permanent - Magnet Motor Drives, Part 1: The Permanent - Magnet Synchronous Motor Drive", 1989, *IEEE Trans. on Industry Applications*, Vol. 25, No. 2, pp 265-273.
8. Richards, R.J., "An Introduction to Dynamics and Control", Longman Group UK Limited, 1979.
9. Sebastian, T. and Slemon, G. R., "Transient Modelling and Performance of Variable-Speed Permanent-Magnet Motors", *IEEE Trans. on Industry Applications*, Vol. 25, No. 1, 1989, pp 101-105.
10. Younkin, G. W., "Modelling Tool Feed Servo Drives Using Simulation Techniques to Predict Performance", *IEEE Transaction on Industry Applications*, March/April 1991, Vol. 27, No. 2, pp 268-274.
11. Zadeh, L. A., "From Circuit Theory to System Theory", 1962, *Proc. IRE*, Vol. 50, pp 856-865.

Modeling of Thermal Analysis of Orbital Spacecraft-Multilayer Insulation Subjected to Hypervelocity Impact Damage

Mohamed A. Gadalla

American University of Sharjah
Mechanical Engineering Department
P.O.Box 26666, Sharjah, UAE
mgadalla@ausharjah.edu

Abstract

This paper presents a computational modeling that predicts the thermal effects associated with hypervelocity impact (HVI) damage to the orbital spacecraft-multilayer insulation (MLI). Every spacecraft that is spending on low-earth-orbit more than several days is subjected to impact of hypervelocity meteoroids and space debris particles (debris of fractured spacecraft and rocket booster stages) that can damage vital systems of the spacecraft. The impact of these particles which can occur at high speeds can degrade and/or damage flight-critical systems and possibly lead to catastrophic failure of spacecraft missions. Therefore, the design of a long duration earth-orbiting spacecraft must consider the effects of such hypervelocity impacts on the subsystems of the vehicle to insure the safety of its occupants and all operating systems. If a spacecraft has a dual-wall system in which a blanket multilayer insulation will be typically included within the wall for thermal protection purposes. A computational model to predict the thermal behavior of MLI was developed. The main goal of this paper is to develop a microcomputer-based design tool to approximately predict the thermal effects associated with the impact damage of the MLI of orbital spacecrafts. The computational model was based on the assumption of axial symmetry about the center of the MLI damage. Finite difference schemes were utilized to discretize the system, where an axially symmetric ring of material was approximately modeled as a single node. It was assumed in the model that the damage in the MLI consisted of a circular hole of the same diameter through all of the MLI layers. Thermal stability of each layer of insulation under steady state conditions was considered in which the heat flux into each node in the system must equal the heat flux out of that node. Since the MLI is in vacuum so the modes of heat transfer are by conduction and radiation. Predictions of the oblique impact model and its effects on the multilayer insulation were discussed and evaluated. This model can be used as a complementary tool in designing a protective structure to prevent such damages and enhance the spacecraft survivability in deep space.

1 Introduction

The existence of orbital debris and its evolution in space is the main concern of all space agencies due to its impact on spacecraft structures. The impacts of these particles can damage flight-critical systems, sensitive electronic equipment, and space structures and possibly lead to catastrophic spacecraft failure. Further creation of many more fragments will be created which can be additional threats to other spacecraft in other orbits. A significant design consideration must take into account the effect of such impacts in the development of earth-orbiting spacecraft. Many researches have been devoted in developing protective shielding systems for spacecrafts and space structures as a means of reducing

such severe damages especially to the flight-critical components (Thomas K., and et al, 2004, Atwell W., and et al, 2004, Pippin, G., 2003, Christiansen, E., and et al., 1999, Schonberg, W., and Ebrahim, A., 1999, Taylor E., and et. al, and Tanaka M., and Moritaka Y., 2004). For unmanned spacecraft, the primary shielding system is provided by the structural subsystem (typically a honeycomb) and the thermal subsystem (multilayer insulation) as indicated by many investigators (Turner R., and et al. and Turner R. and Taylor E., 2001). For surfaces under radiation effects, the honeycomb is typically covered by silverised Teflon tap.

Numerous investigations have been performed to study the effectiveness of multi-wall structures in reducing the damage threat of high speed debris (Schonberg W., and Taylor R., 1990, Schonberg W., 1990, and Schonberg W., and Yang F., 1993). Dual-wall surfaces have proven to provide significant protection against perforation by hypervelocity projectiles. Normal impact as well oblique impact phenomena were implemented by many investigators (Schonberg W., and Mohamed E., 1999, Gehring J., 1970, and Schonberg W., 1989). However, vast majority of tests were performed under normal impact conditions. Recent experimental investigations of high-speed oblique impact have shown that the response of a dual-wall structure is significantly different in its response as compared to a normal impact phenomenon. Additionally, an oblique impact can produce two inward clouds and a tremendous amount of ricochet particles (outward cloud) that could severely damage the external instrumentations carried by the spacecraft. Therefore, oblique impact must be considered during the design stage of a spacecraft structure.

Predicting impact damage associated with hypervelocity orbital debris is a difficult task especially in Earth orbit because of the widely varying conditions of the particles involved. Many analytical models have been developed to predict the response of thin plates to normal and oblique impact phenomena. However, many of these models were developed for relative low impact velocities that are not applicable to hypervelocity impact phenomena. In addition, estimation and analysis of impact damage on space station wall structure using neural networks was implemented by Guleyupoglu and smith (Glueyupoglu S., and Smith R., 1995)]. They derived a set of measures using three-layer backpropagation networks. Two sets of impact damage data were trained. The input parameters for training were pressure, wall thickness, bumper plate thickness, projectile diameter, impact angle, and the projectile velocity. While the output from the neural network consisted of hole dimension for the bumper and the pressure wall, and damage to MLI.

Several researchers and organization such as NASA Johnston Space Center (JSC) and Russian Space Agency (RSA) investigated the feasible methods of cleaning the debris from the Earth-orbit and the associated costs with such methods (Petro A., and Talent D., 1989, and Petro A., and Ashley H., 1989) while others studied the probability of impact damage to spacecraft McCormick B., 1989). Kline, McCaffrey and Stein (Kline R., and et. Al, 1985) suggested possible altitude to the space station not only to minimize the impact damage probability but also minimize the drag and the radiation effects.

Obviously, Space is a hostile thermal environment Thermal instability may exist due to severe solar heat flux. Failure criteria can differ significantly from one location to another on the space vehicle even when the thermal protection system (TPS) material is the same due to thermal effects and mechanical loads encountered by different areas of the spacecraft during mission. To avoid large excursions of temperature, any spacecraft requires some form of thermal insulation. The specification and the characteristics of the required thermal insulation are a function of the spacecraft mission. MLI is mainly used to protect electronic equipment against solar heat radiation. Therefore, it is necessary to investigate thermal effects associated with HVI damage to the space station MLI. The purpose of this study is to

develop a microcomputer-based design model to approximately predict the thermal effects associated with HVI damage to the MLI of a space station.

2 Problem Description

In 1971, Multilayer insulation was first considered as an effective protection shield for interplanetary missions (Howard J., 1969)]. The most accepted configuration for the protection of the pressure wall of a spacecraft or a Space Station involves the use of one or more “sacrificial” bumper layers. This design was first suggested by Whipple (Whipple F., 1947)] and has been adopted by Boeing Aerospace and Electronics. Figure 1 shows the Whipple style of spacecraft pressure wall design that is subjected to oblique impact. In this design, a single bumper layer is placed 100 mm far from the pressure wall. Thirty multilayer insulation of double aluminized Mylar/Dacron netting are located between the bumper and the pressure wall. The bumper layer of the multi-wall structure is intended to protect the pressure wall against perforation by disintegrating or ideally vaporizes any impact orbital debris into several debris clouds. However, three debris clouds are typically formed. Two of them, the normal and in-line debris clouds, travel inward towards the pressure wall. While the third debris cloud, the ricochet debris clouds, travels backward, away from the multilayer insulation. It is hypothesized that the normal clouds contain mainly bumper fragments while the in-line debris clouds could contain mainly projectile fragments. Therefore, three center-of-mass trajectories are associated with the impact damage. Thus, this bumper and MLI in turn insulate the interior of the spacecraft from the cold space environment and reduce the probability of impact damage to the pressure wall. Finally, the main aim of this study is to develop an integrated model between the oblique hypervelocity impact damage to spacecraft and the thermal effects associated with this damage. This include the bumper hole minimum and maximum diameters, multilayer insulation hole diameter, the pressure wall average hole diameter, nodal temperatures of the pressure wall, steady state nodal temperatures of each layer of MLI and the bumper layer, and the amount of water-vapor condensate as the pressure wall temperature drops below the dew point. This condensation could have a hazard effect on electronic equipment on the spacecraft. In this analysis, conservation of mass, momentum, and energy are implemented.

3 Damage Equations

Empirical functions were developed and used by many investigators such as Schonberg and Williamsen (Schonberg W., and Williamson J., 1997) to predict the damage in bumper, MLI, and the pressure wall. For Bumper, MLI, and pressure wall hole diameters can be predicted in the following nondimensional forms:

3.1 Bumper Hole Minimum and Maximum Diameters

The maximum and minimum diameters of the hole can be calculated based on the following equations:

$$\frac{D_{\min}}{D_p} = K_1 \left[\frac{V_p}{V_s} \right]^{K_2} \left[\frac{T_b}{D_p} \right]^{K_3} [\cos \phi]^{K_4} + K_5 \quad (1)$$

$$\frac{D_{\max}}{D_p} = K_6 \left[\frac{V_p}{V_s} \right]^{K_7} \left[\frac{T_b}{D_p} \right]^{K_8} [\cos \phi]^{K_9} + K_{10} \quad (2)$$

3.2 MLI Hole Diameter

$$\frac{D_{MLI}}{D_p} = K_{11} \left[\frac{V_p}{V_s} \right]^{K_{12}} \left[\frac{T_b}{D_p} \right]^{K_{13}} \left[\frac{D_s}{D_p} \right]^{K_{14}} [\cos \phi]^{K_{15}} + K_{16} \quad (3)$$

where D_{MLI} is the average diameter of the hole in the MLI.

3.3 Pressure Wall Effective Hole Diameter:

$$\frac{D_{pw}}{D_p} = K_{17} \left[\frac{V_p}{V_s} \right]^{K_{18}} \left[\frac{T_b}{D_p} \right]^{K_{19}} \left[\frac{D_s}{D_p} \right]^{K_{20}} \left[\frac{T_{pw}}{D_p} \right]^{K_{21}} [\cos \phi]^{K_{22}} + K_{23} \quad (4)$$

where D_{pw} is the effective pressure wall hole diameter.

In Eqs. (1-4), V_p, D_p, ϕ are the velocity, diameter, obliquity of the impacting projectile, while V_s , and K_1, \dots, K_{23} are the speed of sound in the bumper material and function coefficients; respectively.

The coefficients constants were determined using an optimization technique to minimize the errors and match the experimental results as indicated by Guleyupoglu and Smith (Guleyupoglu S., and Smith R, 1995).

4. Thermal Analysis

A computer-based model was developed to predict the thermal behavior of the impact damage of MLI. This model was based on the assumption of MLI axial symmetry. A finite difference scheme was used to discretize the multilayer insulation. The model uses the same number of nodes in each layer. A maximum number of nodes per layer can be obtained to obtain accurate results through refinement. The advantage of this refinement process is that large node spacing is used to calculate the accurate set of nodal temperatures and heat fluxes.

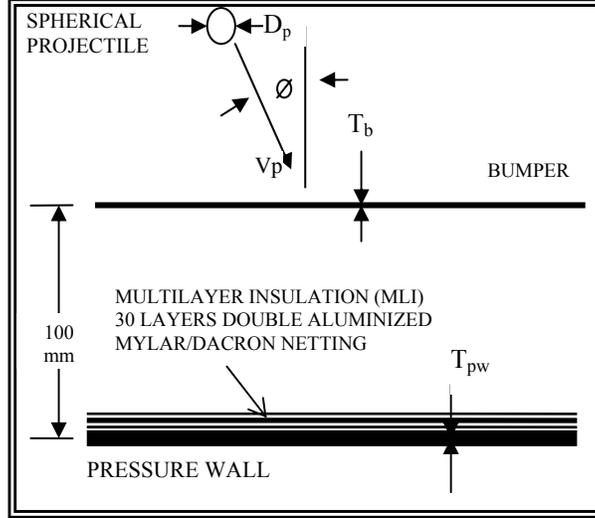


Figure 1: Schematic diagram of generic multilayer bumper wall design

The pressure wall, MLI, and the bumper were assumed radially extended to infinity. Additionally, it was assumed that all MLI consisted of the same number of layers and no lap joints were present in the MLI. It was hypothesized that the damage in the MLI consisted of a circular hole of the same diameter through all of the MLI layers. Each layer of the multilayer insulation was modeled with an array of nodes.

A steady state model was developed based on the amount of heat flux entered each node must equal the heat flux exited from the node. Since the MLI is located between the bumper and the pressure wall in which vacuum exists, the MLI are subjected to conduction and radiation only and the nodal heat flux equations are nonlinear. Basically, the energy equations of the MLI nodes are coupled as well as the nodal temperatures. Figure 2 shows the complex pattern of the heat transfer modes in the MLI zone.

The energy balance on any node on the MLI can be written as:

$$-kt \frac{d^2 T}{dr^2} - kt \frac{dT}{dr} = q_{in} - q_{out} = q_{net} \quad (5)$$

where k is the thermal conductivity of the MLI, r is the radial position of the node, t is the thickness of the MLI, T is the temperature on the node, q_{in} is the heat flux into the layer at position r from adjacent layers, and q_{out} is the heat flux out of the layer at position r from adjacent layers.

Equation (5) was discretized using a finite difference approach to calculate the temperature at each node of the MLI blanket. For the case of a layer with no hole, axial symmetry dictates that the in-plane radial heat flux through the origin ($r=0$) must be zero. The same technique was also utilized for the MLI layers where there was a hole in the insulation in which no radial flux at node 1 because of the existence of the free edge.

In order to treat the outer boundary conditions at the outer edge, the number of layer should be identified in addition to the radius of the area to be modeled. An addition layer would be added on the outer edge of the modeled area to ensure the symmetry in the matrix of the governing equation. It was assumed that the perturbing effects of the MLI hole at the N-th node have diminished. Therefore, the radial heat flux would be neglected and the radial temperature profile is uniform. Thus this boundary

can be modeled by setting T_N and T_{N+1} equal to T_{N-1} . The same scenario would be applicable if the boundary of the modeled area was aligned with the outer edge of the pressure wall, the bumper, and the MLI blanket.

Equation (5) that represents the thermal equilibrium at node i-th can be written as:

$$T_{i-1} \left\{ \frac{kt}{2r_i \Delta r} - \frac{kt}{\Delta^2 r} \right\} + T_{i+1} \left\{ -\frac{kt}{\Delta^2 r} - \frac{kt}{2r_i \Delta r} \right\} = q_{net} \quad (6)$$

The solution procedure for the nodal temperatures started from the pressure wall plate (first layer) and then continued until the nodal temperatures of the bumper plate (final layer). This solution of the nodal temperatures was repeated until the temperatures converge within a prescribed tolerance.

Pressure Wall Nodal Heat Flux

Generally, the pressure wall will interact thermally with the atmosphere inside the spacecraft in which the wall will gain or loose heat depending on the wall temperature as compared to the spacecraft temperature.. This heat transfer mode can be described at i-th node as:

$$q_c = h(T_\infty - T_i) \quad (7)$$

where h is the convective heat transfer coefficient, T_i is the temperature of the i-th node on the pressure wall, and T_∞ is the free stream air temperature inside the spacecraft. The convective heat transfer coefficient can be estimated based on an empirical formula that was presented by Özisik [24] and Bejan [25]:

$$h \approx 4 \left[\frac{u_\infty}{L} \right]^{1/2} \quad (8)$$

where L is the distance traveled by the air along the pressure wall before facing any obstruction, and u_∞ is the air stream velocity next to the pressure wall.

In addition, the pressure wall also radiates heat towards the MLI and into the air stream of the spacecraft module. This heat flux was calculated assuming the emissivities of all surfaces were constant. At the same time, the pressure wall was subjected to heat flux radiated down form adjacent MLI, bumper and space environment if a MLI hole is present. Some of these radiation fluxes were absorbed, emitted, and the others were reflected. For energy balance, the model accounts for all types of radiation. In case of no hole exists in the MLI, all nodal temperatures of the MLI adjacent to the pressure wall will be identical after thermal equilibrium is attained. Therefore, the thermal radiation emitted and reflected will be the same for each node in the MLI layer. In addition, neither thermal energy from space environment nor the bumper will strike the pressure wall.

In more general case when the MLI include a hole, thermal radiation emitted from MLI nodes adjacent to the pressure wall will be different. The thermal radiation emitted from both the space environment and the bumper plate will strike the the pressure wall. Thus, the concept of view factors was considered in estimating the total heat flux by radiation.

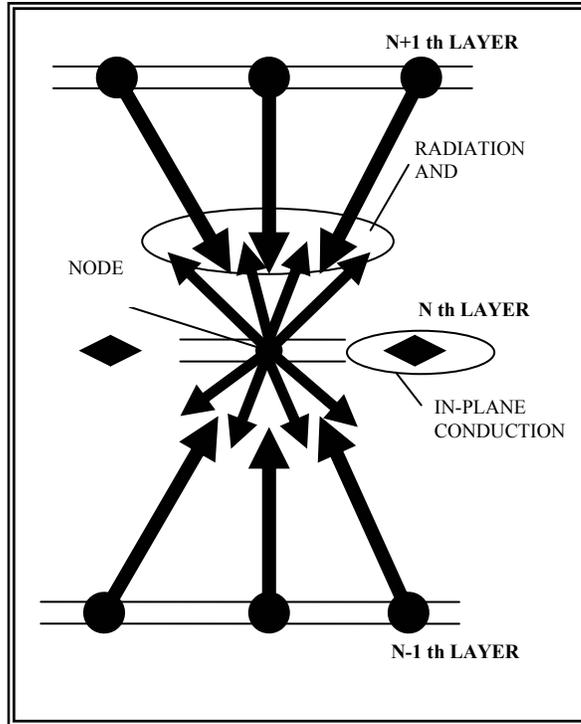


Figure 2. Heat transfer modes in MLI zone

Figure 3 shows the total influx to the i -th node of the pressure wall from the adjacent MLI. The heat flux from the bumper to the pressure wall through the MLI hole was considered in calculating the radiation heat flux. Additionally, the radiation from the space that would pass through the bumper hole to the pressure wall was also considered. This incident solar radiation has an average value of 1353 W/m^2 . This value of the solar gain would be used if the spacecraft is subjected to the sun. But if the spacecraft faces deep space, this solar heat flux will be zero. An average value of 431 W/m^2 would be used if one side of the spacecraft facing sun and the other in on the dark side. Figure 4 shows a schematic diagram that illustrates the input heat flux from the space environment, bumper plate through the MLI hole.

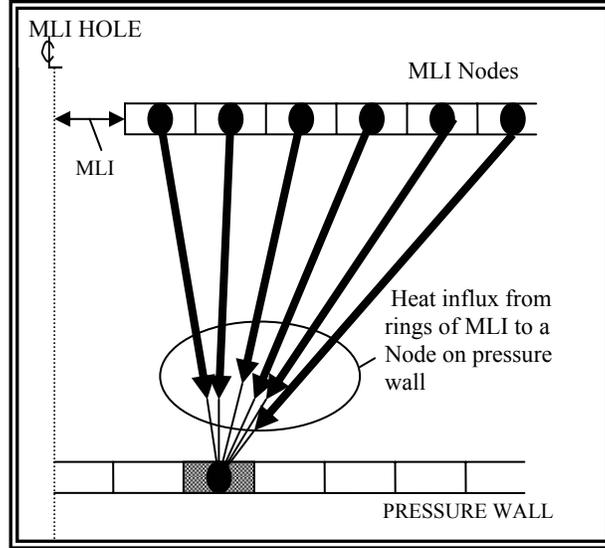


Figure 3. Heat Influx from MLI to a Node in a Pressure Wall

MLI Nodal Net Heat Flux

The MLI next to the pressure wall (first MLI) can radiate energy to both the next MLI layer and the pressure wall. At the same time, the pressure wall would emit and reflect energy to the MLI. This process will be conducted in a reverse manner as shown in Fig. 3. The first MLI will be also subjected to emitted and reflected radiated energy form the adjacent MLI. Therefore, the thermal radiation from the second MLI layer striking the i -th node of the first MLI layer was assumed to equal the radiation emitted from the i -th node of the second MLI layer. In addition, conduction heat transfer from the first MLI layer to the second MLI layer was inhibited by the presence of a layer of Dacron netting. Thus, this heat flux can be described in the following form:

$$q_{Netting} = h_N (T_{i,2} - T_{i,1}) \quad (9)$$

where h_N is the effective netting heat transfer coefficient, $T_{i,2}$ and $T_{i,1}$ are the temperature of the i -th node in the first and second MLI layers, respectively. The netting heat transfer coefficient was assumed the same for all netting layers and equal to $1.0687 \text{ W/m}^2 \cdot \text{K}$.

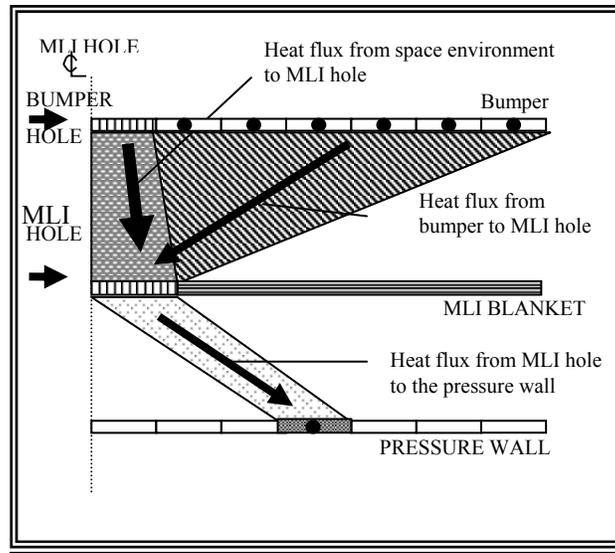


Figure 4: Schematic diagram for the input heat flux from bumper and space environment

Bumper Nodal Heat Flux

The last MLI layer next to the bumper was analyzed as a single layer with the inside surface having the emissivity of aluminized layer and the outside layer having the emissivity of the beta cloth layer. In addition the thermal conductivity of the layer was assumed equal to the weighted average of the inside and outside surfaces. The net heat flux of the combined layer was treated in the same manner as the first MLI layer next to the bumper plate.

The net heat flux to the bumper plate was calculated based on conduction and radiation heat transfer modes since no convective heat transfer from the spacecraft to the bumper plate takes place. Thus, nodal temperatures were calculated layer by layer starting from the pressure wall and ends with the bumper plate. A set of calculation was done globally using an iterative technique until all nodal temperatures including the bumper nodal temperatures converge within the prescribed tolerance. A refinement method for the mesh was adopted until global convergence of all nodal temperatures was obtained.

Condensation Prediction

Condensation of water vapor from moist air is investigated. The main aim is to predict the condensate height for a given temperature distribution on the spacecraft surface. In this analysis, conservation of mass, momentum, and energy are solved with the variation of thermophysical properties due to the changes in temperature. The height of the condensate was determined based on the radial positions of the nodes along the pressure wall, the nodal temperatures, the radius of the surface, and the ambient and dew point temperatures of the moist air and the velocity of the air over the pressure wall surface.

4. Results and Discussion

Figures 5 to 8 show two case-results of the radial pressure wall temperature distributions and the condensate heights that were predicted using the developed computational model. In this model, the following data were used:

Ambient temperature = 294 K

Dew point temperature = 288 K

Coefficient of heat transfer for mist air = $5 \text{ W/m}^2\cdot\text{K}$

Radius of the pressure wall = 1 m

Inlet air velocity = 0.005 m/s.

In the first case, at the MLI hole axis the maximum heat flux is 167.23 W/m^2 , the temperature is 283 K, and the condensation height is 0.0185m. It is found that as the temperature increases, the condensation height decreases. Finally, the results appear to be physically reasonable. While in the second case, the maximum temperature approaches 294 K, and the maximum condensate height is 0.01084 m.

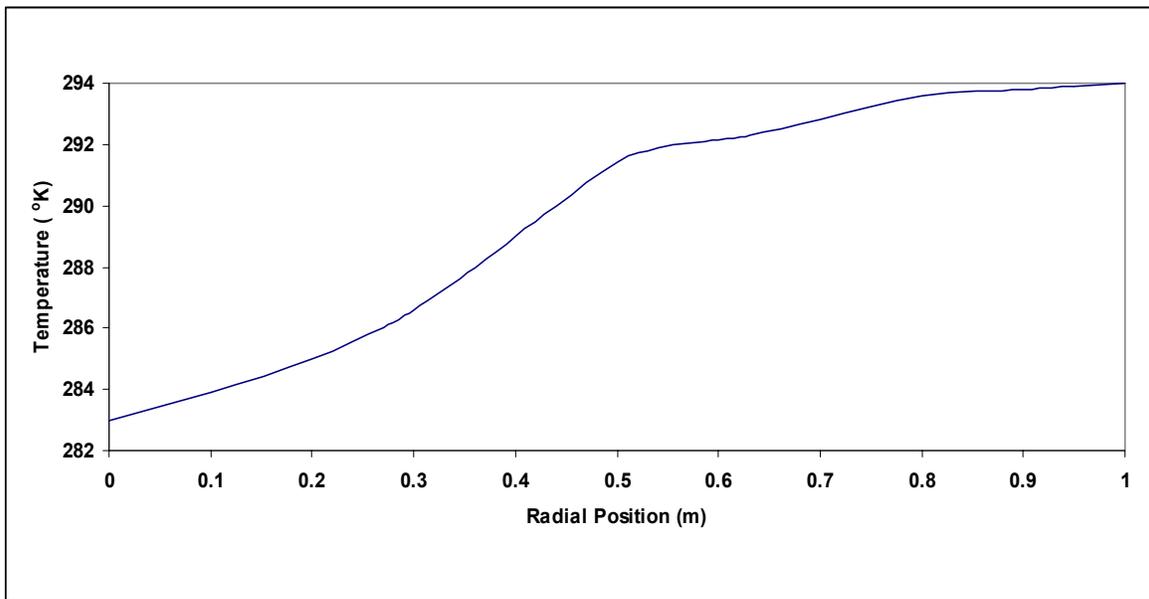


Figure 5: Radial temperature distribution along the pressure wall (case 1)

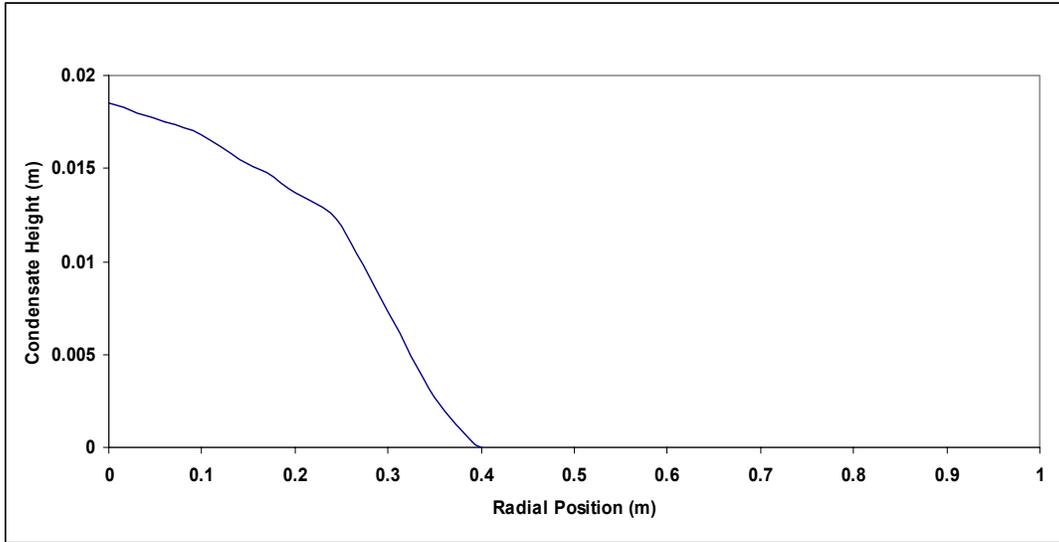


Figure 6: Radial condensate height along the pressure wall (case 1)

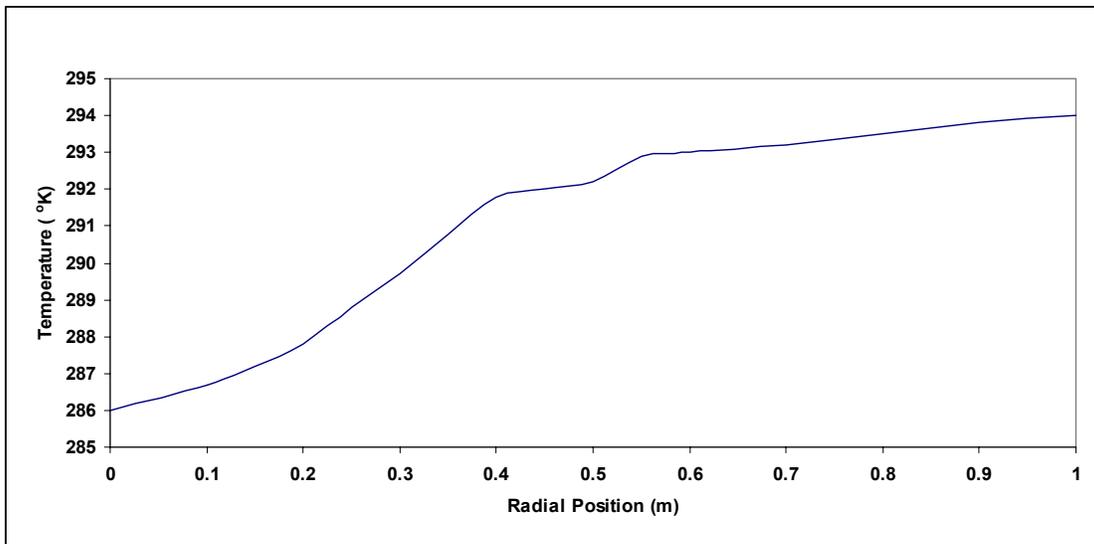


Figure 7: Radial temperature distribution along the pressure wall (case 2)

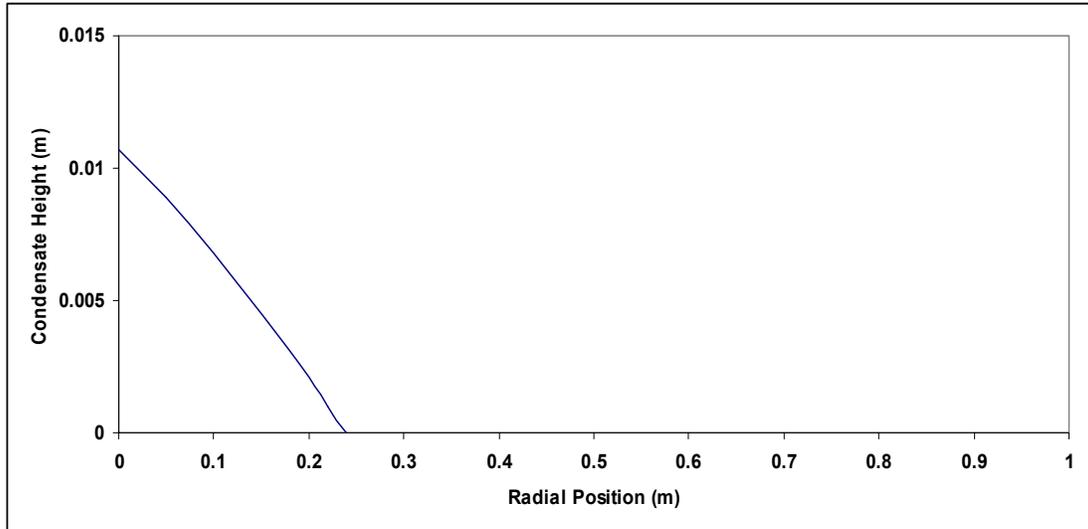


Figure 8: Radial condensate height along the pressure wall (case 2)

5. Conclusions

This paper presents a thermal analysis model that predicts thermal behavior of spacecraft-multilayer insulation subjected to hypervelocity impact. This model calculates all temperature distribution along the pressure wall, bumper plate, and the MLI layers. Condensate height of water vapor was predicted based on the temperature distribution and the environment space temperature. In addition finite difference scheme was utilized to discretize the system and thermal stability of each MLI layer of insulation was considered. This model can be used as a complementary tool in designing the protective shield structure for orbital spacecrafts to prevent hypervelocity impact damage and enhance the spacecraft survivability in deep space. Finally, the model can predict the effect of the bumper temperature on the hole size under different oblique-impact conditions.

Literature:

- [1] Thoma, K., Schäfer, F., Hiermaier, S., and Schneider, E., "An approach to achieve progress in spacecraft shielding", *Ad. Spac Res.* 34: 1063-1075, 2004.
- [2] Atwell, W., Saganti, P., Cuinotta, F.A., and Zeitlin, C.J., "A space radiation shielding model of the Martin radiation environment experiment (MARIE)", *Advances in Space Research*, 33, 2219-2221, 2004.
- [3] Pippin, G., "Space environment and induced damage mechanisms in materials", *Progress in Organic Coatings*, 47, 424-431, 2003.
- [4] Christiansen, E.L., Kerr, J.H., De La Fuente, H.M., and Schneider, W.C., "Flexible and deployable meteoroid/-debris shielding for spacecraft", *Int. J. Impact Eng.*, 23:125-136, 1999b.

- [5] Schonberg, W.P. and Ebrahim, A.R., Modeling of hypervelocity impact phenomena using elementary shock physics, *Int. J. Impact Eng.*, 23, 823-834, 1999.
- [6] Taylor, E.A., Glanville, J.P., Clegg, R.A., and Turner, R.G., "Hypervelocity impact on spacecraft honeycomb: hydrocode simulation and damage laws", *Int. J. Impact Eng.*, 29, 691-702, 2003.
- [7] Tanaka, M. and Moritaka, Y., "Single bumper shield based on vectran fibers", *Advances in Space Research*, 34, 1076-1079, 2004.
- [8] Turner, R.J., Taylor, E.A., McDonnell, J.A., Stokes H., Marriott, P., Wilkinson, J., Catling, D.J., Vignjevic, R., Berthoud, L., and Lambert M."Cost effective honeycomb and multi-layer insulation debris shields", *Int. J. Impact Eng.*, 26, 785-796, 2001.
- [9] Turner, R. and Taylor, E.A., "Cost effective debris shields for unmanned spacecraft (Final report), ESTEC contract 12378/97/NL, March 2000.
- [10] Schonberg, W.P. and Taylor, R.A., Exterior spacecraft subsystem protective shielding analysis and design", *J. Spacecraft and Rockets*, 27, 267-274, 1990.
- [11] Schonberg, W.P., "Hypervelocity impact penetration phenomena in aluminum space structures ", *J. Aerospace Eng.*, 3, 173-185, 1990.
- [12] Schonberg, W.P. and Yang, F., "Response of space structures to orbital debris particle impact", *Int. J. Impact Eng.*, 14, 647-658, 1993.
- [13] Schonberg, W.P. and Mohamed, Essam, Analytical hole diameter and crack length models for multi-wall systems under hypervelocity projectile impact", *Int. J. Impact Eng.*, 23, 835-846, 1999.
- [14] Gehring, J.W., Engineering considerations in hypervelocity impact, in: Kinslow, R. (Ed.), *High-Velocity Impact Phenomena*, Academic Press, New York, pp. 463-514, 1970.
- [15] Schonberg, W.P., "Penetration and ricochet phenomena in oblique hypervelocity impact", *AIAA Journal*, 27, 639-646, 1989.
- [16] Glueyupoglu, S. and Smith, R.E., Prediction of hypervelocity orbital debris impact damage to the space station by neural networks", *Math. Comput. Modelling*, 21, 229-242, 1995.
- [17] Petro, A.J. and Talent, D.L., "Removal of orbital debris, In *Orbital Debris from Upper Stage Breakup*, Loftus, J.P. (Ed.), 121, 169-182, AIAA, New York 1989.
- [18] Petro, A.J. and Ashley, H., "Cost estimates for removal of orbital debris, In *Orbital Debris from Upper Stage Breakup*, Loftus, J.P. (Ed.), 121, 183-186, AIAA, New York, 1989.
- [19] McCormick, B., "Collision probabilities in geosynchronous orbit and techniques to control the environment", In *Orbital Debris from Upper Stage Breakup*, Loftus, J.P. (Ed.), 121, 187-197, AIAA, New York, 1989.
- [20] Kline, R., McCaffrey, R., and Stein, D.B., „A summary of potential designs of space stations and platforms, In *Space Stations and Space Platforms-Concepts, Design, Infrastructure, and Uses*, Bekey I., and Herman, D. (Ed.), 99, AIAA, 1985.
- [21] Howard, J.R., "Development of the Marine Mars 1971 meteoroid shield", *AIAA Hypervelocity Impact Conference*, Cincinnati, Ohio, pp. 569-377, 1969.
- [22] Whipple, F.L., "Meteorites and space travel, *Astronomical Journal*, 131, 1947.
- [23] Schonberg, W.P. and Williamsen, J.E., "Empirical hole size and crack length models for dual-wall systems under hypervelocity projectile impact", *Int. J. Impact Eng.*, 20, 711-722, 1997.
- [24] Özisik, M.N., *Heat Transfer: A Basic Approach*, McGraw Hill Book Company, New York, 1985.

A self organized holonic control for mechatronics complex systems: application to a robotized car park

Patrick Pujo, Fouzia Ounnar, Cecilia Zanni

Université Paul Cézanne

Laboratoire des Sciences de l'Information et des Systèmes - UMR CNRS 6168

Avenue Escadrille Normandie Niemen, 13397 Marseille cedex 20, France

{patrick.pujo, fouzia.ounnar, cecilia.zanni}@lisis.org

Received May 28, 2005

Abstract: *In this paper, we describe the two conditions so that a complex system, composed of several mechatronics machines, may be qualified as a mechatronics system. The first condition reflects the aptitude of these machines to work together without the intervention of a central decision system of higher hierarchical level. The second reflects their aptitude to manage their own behavior and to generate the tasks to be carried out in the context of their execution. A self organized holonic control accomplishes these conditions. An example of implementation of this approach is presented with the application to a robotized car park.*

1 Introduction

We can define a mechatronics object as being the result of the integration of elementary mechanical and electronic components. The integration means the localized association of these components: the mechatronics product is more compact. It implies that the mechatronics product seems more complex due to the synergetic tangling of those two technologies. In return, it is more effective than a traditional product. In particular, electronics brings to the mechanical engineering all the possibilities of intelligent control. This intelligent control supports numerous and new functions, such as surveillance, diagnosis, communication...

Can a complex system, composed of several mechatronics machines, be itself described as a mechatronics system? Our answer is positive when the component machines are able, in an autonomous and coordinated way, to ensure its operation, without the intervention of a central supervision, command or management system. We speak then about self organized control functions, which are integrated in the intelligent control of each mechatronics machine. We will describe a holarchic approach of such a solution, where the control of each mechatronics machine can be easily equipped with these two particular functions. The first makes it possible to manage the interactions with the other mechatronics machines. The second makes it possible to place the capacities of the machine with respect to what the other machines can do so that the total system can continue to evolve.

After having presented the various conceptual proposals for our approach, we will develop an example of application.

2 Self Organized Holonic Control for Mechatronic Complex Systems

The organization of the decision-making system configures the ability of a complex system for being a mechatronics one.

2.1 Self Organized Holonic Control

The concept of organization must firstly be specified to define the one of self-organization. A production system is organized when each one of its constitutive entities has a completely formalized behavior. This behavior only depends on the stimuli coming from external orders emitted by a supervisor that manages the coordination. This classical approach of decision-making is naturally strongly centralized: the supervisor generates at level N+1 the group of necessary orders to set in motion the entities of level N.

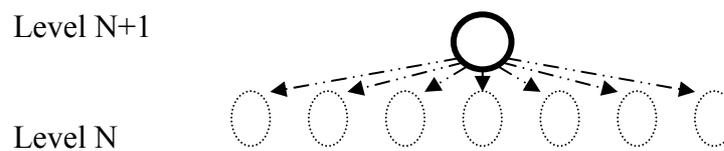


Figure 1: Centralized organization of the decision-making

In the world project IMS (Intelligent Manufacturing System) (*IMS, 1997*), the concept of self organized system is defined as a "system not-coordinated by outside. The elements are endowed with autonomy and carry out tasks together, in interaction and mutual comprehension; the sum, or combination, of the individual tasks allows to generate an order, a good or a more total service". The concept of self-organization is defined as the process during which structures emerge at the collective level (appearance of a structure on the N+1 scale starting from a dynamics defined on the N scale) starting from a multitude of interactions among individuals.

Three remarks underline the link between autonomy of each decision-making center and its aptitude for self-organization.

Firstly, we must regard the self-organization as a mode of decision-making in real time without preliminary estimated organization. Indeed, if there were a previously organization (with classical control), there would not need more to be self-organized (with intelligent – or mechatronics – control). This implies a temporal horizon of decision-making in a very short term, because the events occurring at every moment can involve brutal ruptures of behavior. Then, to get an organization, one needs a common goal to these decision-making centers. We can get different versions of this organization, according to the machine properties and the tasks characteristics that we want to organize. For the tasks to be executed, this is translated into terms of synchronization, coordination, cooperation, negotiation and/or generation. A holon is a good concept to represent a decision-making center (*Koestler, A, 1989*).

The solution finally adopted to make the set of these components operational will be obtained by emergence. Indeed, when there is no hierarchy, each component is involved, on the one hand in the proposal for solutions, and on the other hand in the evaluation of solutions. The most powerful proposal from the point of view of the evaluation criteria is the one to be adopted.

A flat holonic form (Fig 2) largely facilitates its implementation (*Bongaerts et al., 2000*). Indeed, on the one hand, if the self-organization of many entities could be gathered in a decision-making center, it would be necessary to put there all the necessary information relative to the state of each of the entities. Information flow would be then complex to

implement, from the point of view of its quantity and reliability. On the other hand, the localization of processing on the components themselves, nearest to the necessary information, is very interesting, with one communication level.

We define a self organized holonic control in the following way: the decision system of a set of holons $\{H_i\}$ is structured in a flat form where these holons jointly ensure the decisions concerning themselves, without instruction or order coming from the decision-making center of higher level and thanks to functional primitives duplicated on each one of them and in interaction via a common communication protocol.

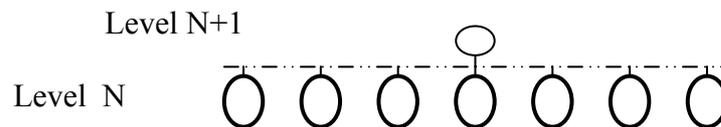


Figure 2: Flat holonic form of the decision-making

This concept of interaction induces the concept of protocol, whose principal function is to specify the organization rules, i.e. the set of potential relations between the system components. Then, that frames a control structure that can be represented into a physical structure and achievable actions by the corresponding equipment.

2.2 Interaction Protocol between Holons

This decision-making mechanism, standard functions of each holon, is based on the competition of the same participant machines according to a Call For Proposals (CFP) launched by an initiator holon. This mechanism is repeated for the execution of each task (*Broissin, N., 1999*). When an initiator holon H_λ has to execute a new task, the task organization problem consists in determining the holon which will execute it. This holon H_λ broadcasts a Call For Proposals to the others by means of the communication protocol. Each participant holon listens to the message and then uses its task generation function to determine if it can assign this task to itself. In fact, each holon estimates a performance, which depends on common criteria among all participant holons. To estimate this performance, each holon has its own task generation function, which simulates and evaluates the task, taking its load state into account. The holon H_λ compares the result of this simulation with a possible result already released by other participant holons. If its result is better than the released one, then the entity releases its result and then commits itself to execute the future task, while another entity does not release a better result (example: Fig 3).

This task assignment process is inspired by the Contract Net Protocol (CNP) (*Smith, R.G., 1980*). In classical hierarchical architecture, five basic holons (product, machine, scheduler, computing and negotiation) cooperate as equals (*Kanchanasevee et al., 1997*). In CNP, the initiator sends out a Call For Proposals. Each participant reviews CFP's and bids on the feasible ones accordingly. The initiator chooses the best bid and awards the contract to the considered participant. Finally, the initiator rejects the other bids. Each initiator thus ensures the management of the CFP which it sent, in a temporarily centralized way.

Compared to this basic operation, we propose several contributions that come to enrich this protocol and simplify it. Indeed, the idea is to minimize the number of interactions and messages, and to remove any risk of blocking in the case of disturbances in the communication system among holons. The CFP is always launched by the initiator for all the participants, with a deadline. Each participant listens to all the messages concerning this CFP and builds its answer according to the contents of these messages. He answers as soon as he can provide an answer, and only if its proposal is better than those already sent on the

network. If it cannot provide an answer containing a better proposal before the deadline, it never answers. At deadline, all participants and the initiator know the participant who has won the contract and who must perform the task.

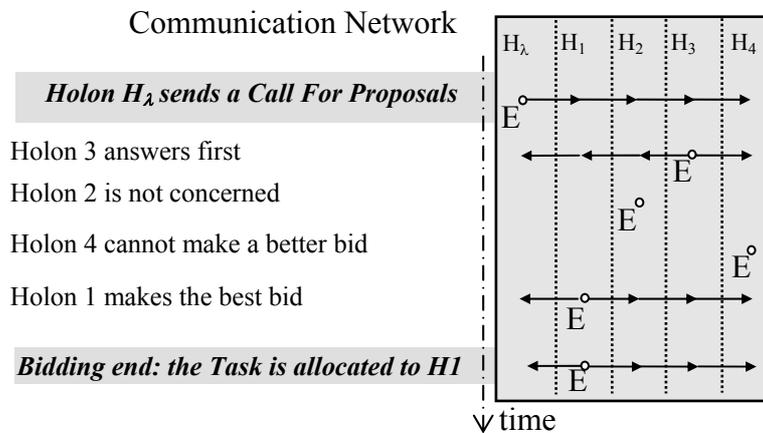


Figure 3: Interaction protocol between holons

The Foundation for Intelligent Physical Agents representation (FIPA, 2002) of this new CNP form, called Mechatronics Interaction Protocol (MIP), is shown in Fig 4.

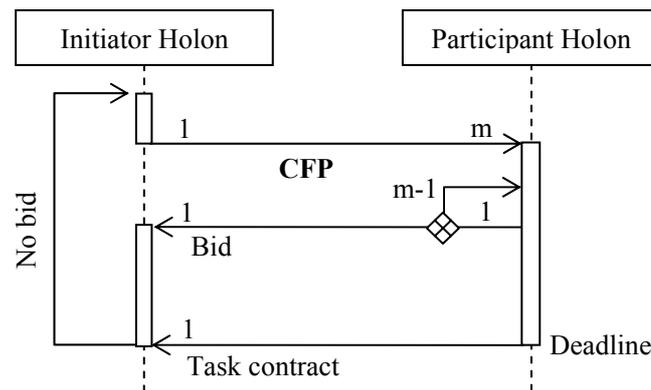


Figure 4: FIPA representation on MIP

If no bid is proposed before the deadline, the CFP must be started again, after, possibly, a modification. Every holon can be an initiator or a participant. Several CFPs are simultaneously in negotiation, since they correspond to tasks to be carried out in the near future on the machines. A holon can be thus simultaneously initiator of a CFP and participant of several others. This interaction protocol gives a strong dynamics to the system control. It confers to the holon an intelligent behavior with respect to the total operation of the system to which it belongs. It is for this reason that we can qualify the holon and its associated machine as ' mechatronics '.

All this decision-making mechanism depends on the capacity of each holon to propose an evaluation of its own performance to carry out a task to come. This is the object of our next paragraph.

2.3 Evaluation of local performance by a task generation function

For each task described in its CFP, the evaluation of each machine performance must hold account of the machine possibilities and the task execution conditions. The task generation function allows the creation of order programs corresponding to the execution of these tasks. These programs can be evaluated. We are thus here intrinsically related to the production function of the machine. For each new type of machine, it is thus necessary to make a fine analysis of its capacities, in order to obtain a model of its own potentially realizable operations. This set constitutes the expertise of the machine in the form of parameterized operations. These operations models are founded on a technological decomposition of the production tasks and their environment, associated with automatic generation algorithms of chronologic actions and elementary trajectories. If the knowledge related to these two aspects may be formalized, then any mechatronics machine with these characteristics can be equipped with a task generation function.

This task generation function is requested twice. The first request is started by the interaction protocol module, in order to give an estimate of the performance with respect to a task being the object of a CFP. The second request takes place once that the task is definitively assigned to the machine, in order to generate the operating sequence which will be actually carried out by it.

When this function is associated with a NC machine tool, associated knowledge concerns the metal-cutting. For example, in turning, the expertise can be formalized in an algorithmic form. From description of a turning task (*Pujo et al., 1996*), characteristics of a given NC lathe and the state of its tools, the generation of the tool trajectories by the generation function associated with this lathe will proceed according to following steps: choice of the conditions for catching the piece on the lathe, choice of the tools for each operation, choice of the cutting conditions for each operation, generation of the trajectories of tools and translation into a task program for Numerical Control (ISO language). These algorithms, traditional in CAM, and the associated expertise are described in (*Anselmetti, B., 1990*). To calculate the obtained performance corresponds to calculate the cost of a transformation by machining. It is necessary to include the cost of the machining itself (which depends on the hourly cost of the machine and the duration of the machining, which itself depends on the continuous paths generated, the power of the machine, the cutting conditions selected...), the cost of the tools (which wear out), the cost of machine setup (which is in fact the cost of immobilization of the resource for a proper configuration), the cost of the raw material, and the cost of latencies during loading and unloading a piece (*Broissin, N., 1999*). This performance is thus used as a comparison indicator of the various lathes in a robotized turning cell.

When the generation function of the tasks to be carried out on a machine is function of the current or foreseen tasks on the other machines, it is necessary to take into account the evolution context of the system as a whole. The task generation is carried out according to the present state of each machine and their foreseen tasks in the short term. That is the case when we encounter problems of potential collisions among machines: AGV control problems (*Broissin et al., 1996*) or automated transit rail hub. We will meet this type of generation function in the following example.

3 Application Case: a Robotized Cars Park

The various concepts which we have just presented have to be tested and validated before an implementation on real systems. With that goal, we develop next the simulation model based on the example of a robotized cars park.

3.1 Presentation of a Robotized Cars Park

In the heart of the great urban centers, there are few sites to establish car parks. Moreover, the traditional car parks, where the driver leads his vehicle to the parking place, potentially generate insecurity: robbery, aggression...

An answer can come from the establishment of robotized car parks in the town centers, where the users do not get in and that to equal volume, can store up to twice more vehicles (the places are narrower, the floors are smaller and the circulation spaces occupy fewer surfaces). The architecture of this type of car park is structured around central circulation alleys for mobile robots (Fig 5). On both sides of these alleys there are storage sections, organized in places which can contain one vehicle each. On the same floor, the storage section is composed by two lines of places. Several planes of storage may be superposed, in the air (building) and/or invisible (underground car park). The access to each place is done thanks to a mobile robot, transporting the car according to direction X and circulating between the two lines of places. The passage from one floor to another is done thanks to elevators (direction Z) at each end of the alleys.

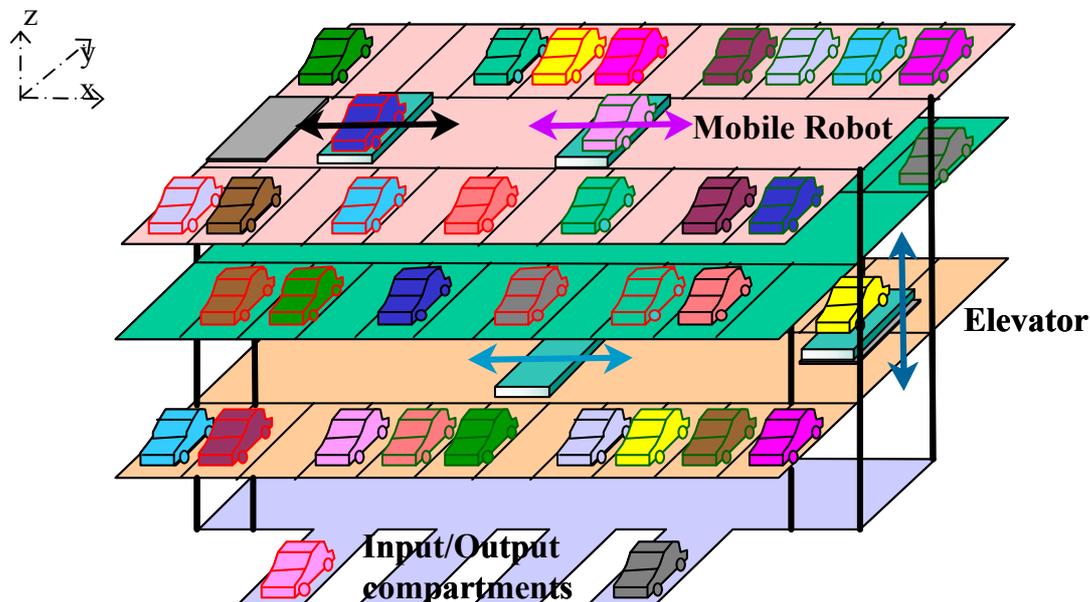


Figure 5: Schematic View of Robotized Cars Park

Input/Output compartments are located at the street level and allow the loading and the unloading of the mobile robots, i.e. the deposit and the recovery of the vehicles by their users. Each mobile robot must provide the following functions:

- to travel in the alleys, to be able to position themselves in front of the places and to get in/out the elevator robots,
- to take or leave a vehicle in a place or an I/O compartment, independently of the dimensions and other characteristics.

A mobile robot, then, consists of two parts. A carrier ensures circulation in the alleys and the access to the elevators (movement according to axis X). This last allows the fine positioning of a unit 'shuttle & vehicle' opposite the parking places (Fig 6). A shuttle ensures the gripping of the vehicle and can be detached from the carrier to get in the places and to take or leave the vehicle there (movement according to the axis y).

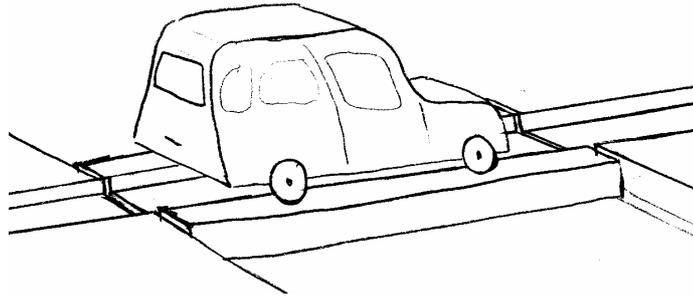


Figure 6: Mobile Robot, facing two places

The shuttle circulates on the carrier and in the places. It passes under the vehicle to seize it by its wheels. It is telescopic to adapt to the distance between the axles of the vehicle.

To arrive to this result, the on-board technology in each mobile robot (MR_i) is highly complex and integrated. The other machines (Elevator Robots (ER_k) and Input/Output Compartments (IOC_z)) can also be regarded as mechatronics machines. Our contribution consists in equipping them with all the sufficient autonomy so that they do not require a central command system, and that the complete car park may be considered as a mechatronics system.

To do so, there will be a communicating computing unit on each machine that will support the holon algorithms: the interaction protocol and the task generation function. It is the latter that will be described next. We will focus on the operation of the mobile robots, which is the most complex.

3.2 Mobile Robot Task Generation

Each MR has to generate its trajectories alone. For that, it has a data structure representing the car park. It knows also the last known position of the other machines, and their estimated movement plan for the current tasks. At time t when it receives the CFP, it has to look for the best possible path. It thus has to be in front of the starting point at the good moment and then it has to reach the arrival point as quickly as possible, to be available again for a new mission. The particular architecture of the car park (an elevator at each end of the alleys) makes the number of possible paths not too high. Unfortunately, these paths are also occupied by other mobile robots, which constitute mobile obstacles to avoid. However, these obstacles are temporary, and it can thus be advisable for the MR to stop waiting for them to disappear. Afterwards, these paths go by the elevator robots, which require again synchronization and waiting. The path taken by the MR must thus be associated to its temporal component to build the MR trajectory. The selected trajectory will be the one with the earliest completion date.

For each segment of the path, the initial occupation appears, and therefore the non availability of the segments, as a function of time. This path is analyzed in the plane (x,z) , with remarkable positions of the MR, facing the parking places. We will note $p_{k,i}$ this position, with k the number of the floor and i the index of the corresponding place.

Let us take the example, Fig 7, of the task associated to a car parking. The parking takes place in an input compartment located in position $p_{1,4}$. A free place was reserved in position $p_{5,12}$. The task generation thus consists in finding the best trajectory to go from position $p_{1,4}$ to position $p_{5,12}$, taking into account the above mentioned constraints. We notice that the 5th floor is already occupied by the MR_α , who arrives in the left elevator, it joins position $p_{5,12}$ to leave a vehicle, then moves towards position $p_{5,8}$ to take a vehicle and take it towards the right elevator to, finally, go down again towards the output compartment.

To answer the CFP, the MR_α holon has several solutions to evaluate: going by the left elevator, going by the right elevator or combining both while crossing an intermediate floor. There may also be obstacles on all the possible paths and the task may become impossible for it. We will examine only the first solution.

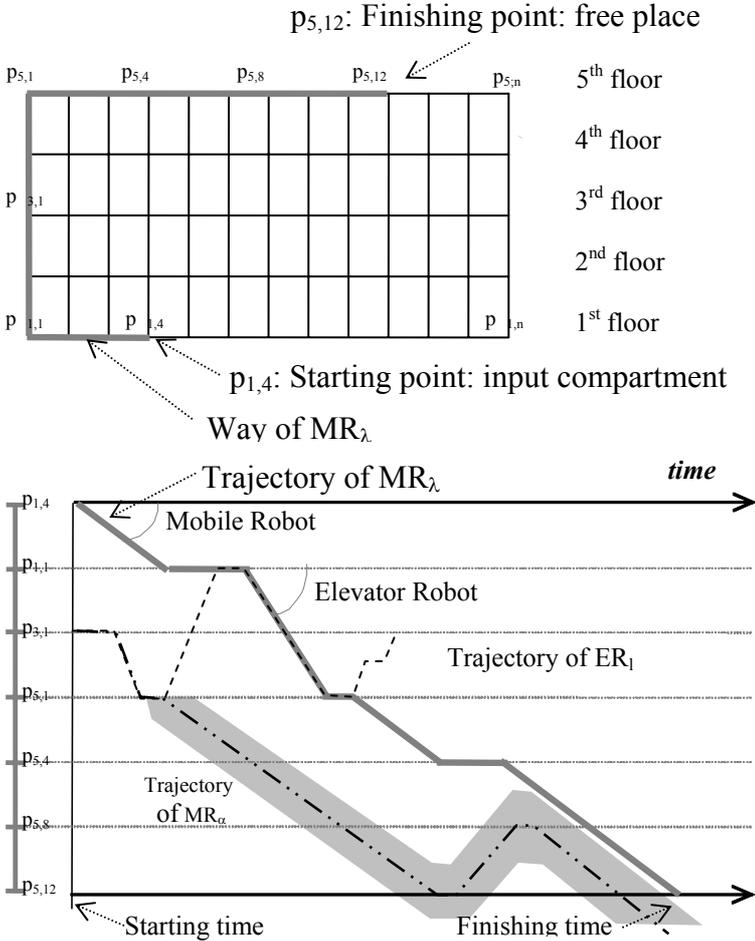


Figure 7: Schematic View of MR trajectories

MR_λ leaves position $p_{1,4}$ and moves towards the left elevator (position $p_{1,1}$). There, it waits for the arrival of the elevator which must be free. For that, it already checked, by a CFP, the availability of the elevators. Once in the elevator, it is brought to the 5th floor, where it gets out (position $p_{5,1}$).

From there, it arrives to the waiting position $p_{5,4}$, so that MR_α completes its operations. As soon as MR_α releases the path, MR_λ follows it by keeping an anti-collision safety distance with it.

Finally, the finishing time of this solution is easy to evaluate. It constitutes the performance of this solution. Each MR will test and evaluate all the solutions relating to it, and will retain the best. It will compare then, via the interaction protocol, this performance with those obtained by the other MR. The most efficient MR will be retained.

3.3 HLA Simulation for MIP Control Validation

For rapidly implementing a flat holonic form, we have used the HLA framework (*IEEE P1516; IEEE P1516.1; IEEE P1516.2*) that is the standard for distributed systems simulation.

The High Level Architecture (HLA) is a software architecture for creating computer simulations out of component simulations. The HLA provides a general framework within which simulation developers can structure and describe their simulation applications.

The HLA was developed by the Defense Modeling and Simulation Office (DMSO) of the Department of Defense (DoD) to meet the needs of defense-related projects, but it is now increasingly being used in other application areas. Examples of non-military applications that have already used the HLA are traffic simulations and factory production line simulations.

We can consider a complex simulation as a hierarchy of components of increasing levels of aggregation. At the lowest level is the model of a system component. This may be a mathematical model, a discrete-event queuing model, a rule-based model, etc. The model is implemented in software to produce a simulation. When this simulation is implemented as part of an HLA-compliant simulation, it is referred to as a federate. HLA simulations are made up of a number of HLA federates and are called federations.

The HLA consists of three components: HLA rules, interface specification and Object Model Template (OMT).

At the highest level, the HLA consists of a set of ten HLA rules which must be obeyed if a federate or federation is to be regarded as HLA-compliant. The HLA rules are divided into two groups consisting of five rules for HLA federations and five rules for HLA federates.

The interface specification defines the functional interfaces between federates and the runtime infrastructure (RTI). The RTI is software that conforms to the specification but is not itself part of the specification. It provides the software services, which are necessary to support an HLA-compliant simulation.

The interface specification identifies how federates will interact with the federation and, ultimately, with one another (Fig 8).

Reusability and interoperability require that all objects and interactions managed by a federate and visible outside the federate should be specified in detail with a common format.

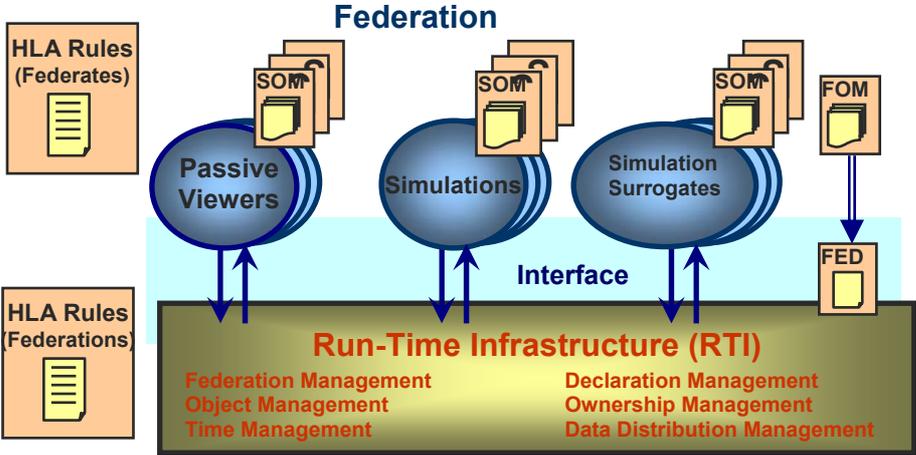


Figure 8: HLA structure

The Object Model Template (OMT) provides a standard for documenting HLA Object Model information. In our simulation environment, every holon is a federate. There are also mobile robot federates, elevator federates and input/output compartment federates. This allows correctly directing the CFP to the correct participants. All these federates communicate with each other via the RTI, over the common interface base provided the interaction protocol.

3.4 Realization

Practically, a PC network allows, then, to test and validate our flat holonic form. Once the algorithms validated, it is necessary, of course, to implement them on the on-board hardware of the machines. What is important to remark is that either by simulation or on-board, the algorithms will remain the same.

Fig 9 shows an UML Sequence Diagram example, for the following situation: A mobile robot going from the input/output compartment in the same floor to a place in a different floor.

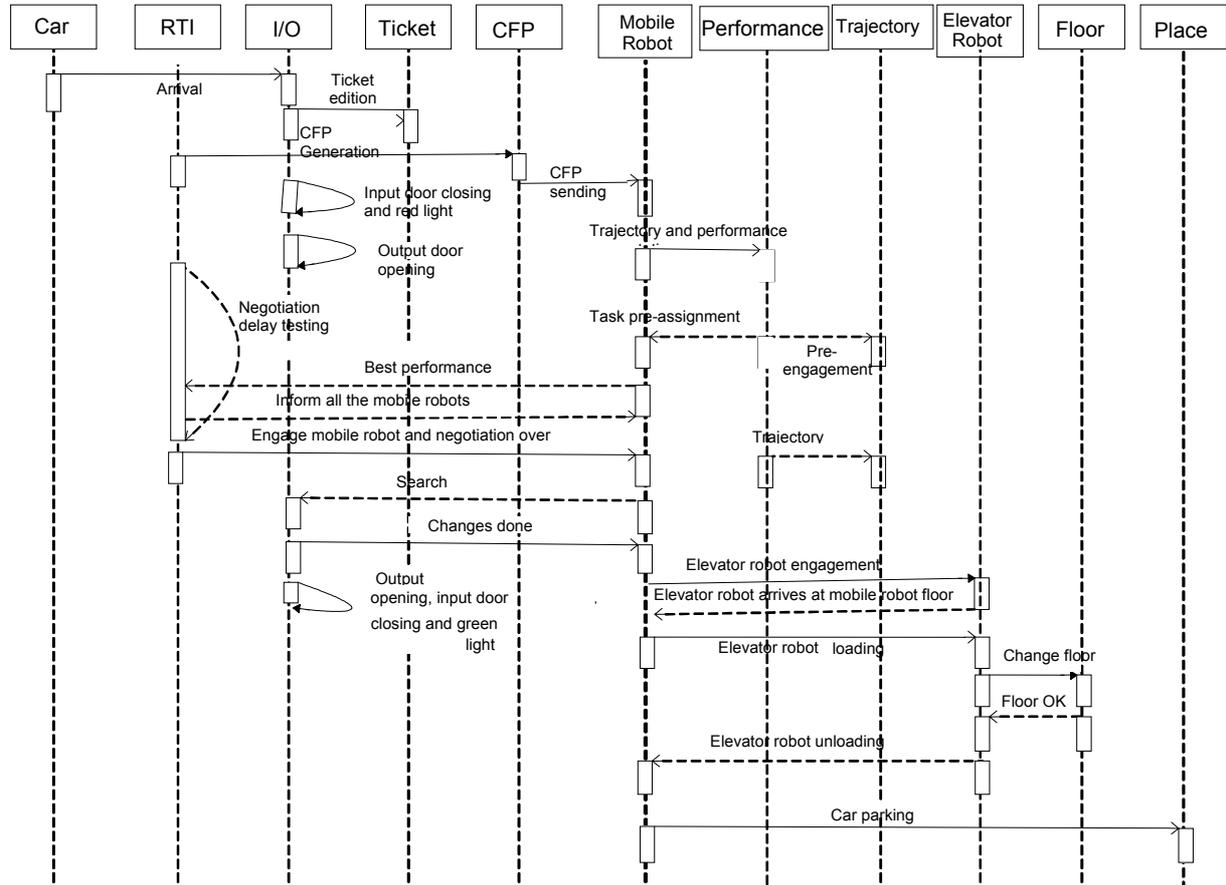


Figure 9: Sequence diagram

The steps presented in this sequence diagram are:

1. The client leaves his car in the I/O compartment.
 2. The client presses on the desk key at the I/O compartment, for his car being taken care by the automated car park, the desk prints his park ticket.
 3. A CFP for car parking is then generated.
 4. This CFP is sent via the RTI to all the mobile robots in the park.
 5. The input door of the compartment is then closed and the light turns to red, to indicate that the car is being taken care.
 6. The output door of the compartment opens to allow the mobile robot load the car to park.
- As long as the CFP negotiation delay is not finished*
7. Every mobile robot calls for its performance module to compute the trajectory and performance for the CFP that it is taking over.
 8. If the computed performance is better than the one that is booked at that moment, then this mobile robot will be pre-assigned the task and it will also pre-engage the segments of the pre-assigned trajectory.

9. The mobile robot that has been pre-assigned the CFP will inform all the others the new performance and the CFP pre-assignment.

CFP negotiation time over

10. Once the CFP negotiation time is over, that mobile robot effectively engages on the task.

11. The mobile robot engages the set of segments that constitute its trajectory.

12. The mobile robot moves horizontally up to the I/O compartment.

13. The mobile robot loads the car that is in the I/O compartment.

14. The output door of the compartment closes, the input door of the compartment opens, the light turns to green.

Since the mobile robot is not at the same floor of the parking place, it has to engage an elevator robot

15. The mobile robot asks the elevator robot to come and search it at the floor where he is at that moment.

16. The elevator robot indicates its arrival to the mobile robot.

17. The mobile robot loads into the elevator robot.

18. The elevator robot, loaded with the mobile robot, goes up or down, to the corresponding floor.

19. The elevator robot reaches the floor where there is the parking place.

20. The elevator robot unloads the mobile robot.

21. The mobile robot moves to the parking place to leave the car.

The evolution of the various robots can be graphically observed via Windows. This interface is also useful for manual controlling (vehicle input/output by click on buttons) or automatic (choice of the frequencies of input/output, on average and standard deviation) and makes it possible to visualize indicators of performance.

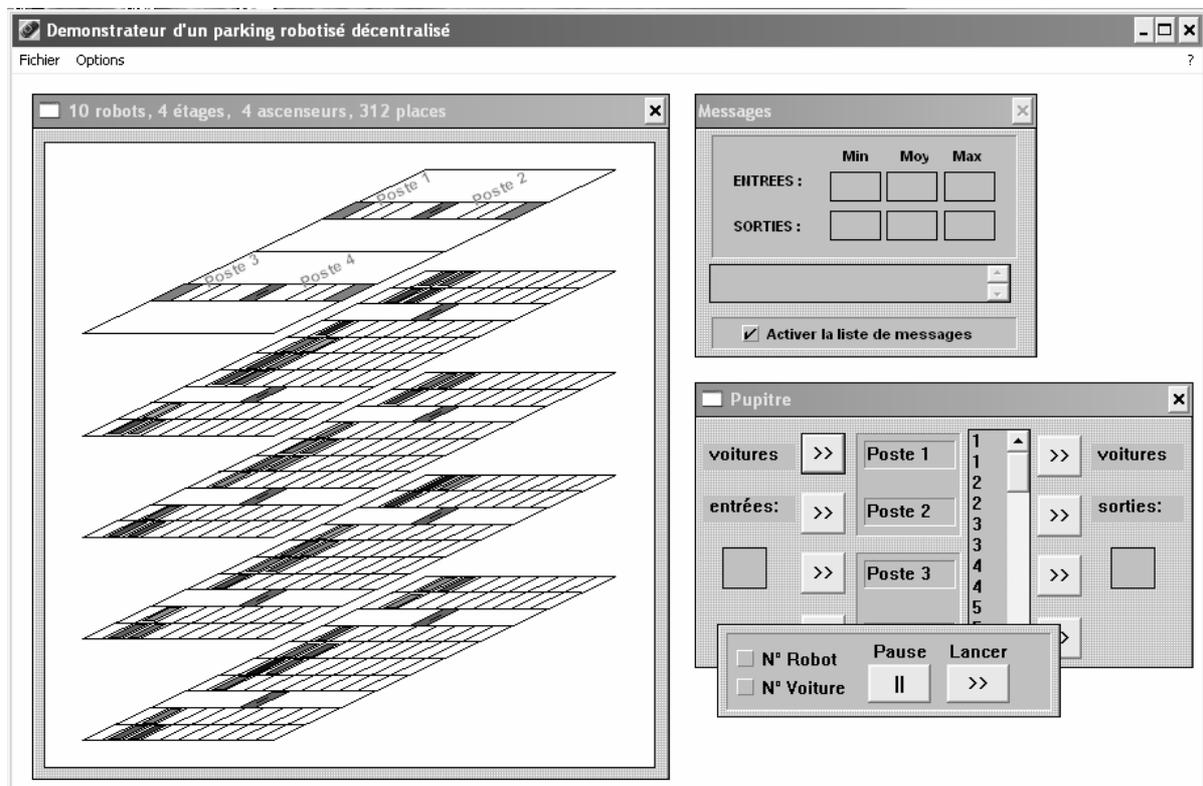


Figure 10: Display window of the simulation model

4 Conclusions

In this paper, we have seen that we can characterize a complex system of 'mechatronics' when we can confer to each one of its components the sufficient decisional capacity, for its own behavior and for the collective behavior. This approach allows, without having to implement centralized systems for scheduling and trajectory planning, to obtain the optimized trajectories, i.e. the shortest possible. This makes it possible to minimize the waiting of customers, who have, in this way, a better quality of service. This approach can be applied to various applications (*Broissin, N., 1999; Pujo et al., 1996; Broissin et al., 1996*), that are equivalent to systems mechatronics composed of mechatronics machines. This approach represents a strong potential on research and development for automated and robotized systems.

5 References

- Anselmetti, B., "Algorithme de génération automatique d'une phase de tournage", *La gamme automatique en usinage*, groupe GAMA, Hermès, 1990
- Bongaerts, L., Monostori, L., McFarlane, D., and Kadar, B., "Hierarchy in distributed shop floor control", *Computers In Industry*, vol 43, 2000, pp 123-137.
- Broissin N., *Contribution à l'amélioration de la réactivité des systèmes de production automatisés et flexibles grâce à un pilotage basé sur une génération de tâches décentralisée*, PhD en Sciences, Université d'Aix-Marseille III, 19 juillet 1999.
- Broissin, N., Pujo, P. and Bertrand, J.C., "Decentralized piloting of transport system by autonomous vehicles", *WAC 96 - World Automation Congress - ISRAM 96*, pp 125-130, Montpellier, 27 - 30 mai 1996
- FIPA, "FIPA Contract Net Interaction Protocol Specification", *Foundation for Intelligent Physical Agents*, December 3, 2002.
- IEEE P1516 Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules.
- IEEE P1516.1, Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Federate Interface Specification.
- IEEE P1516.2, Draft Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Object Model Template (OMT) Specification.
- IMS, 4th IFAC WS on Intelligent Manufacturing Systems, *IMS'97, July 21-23, Seoul, Korea - Proceedings edited by Jongwon K., PERGAMON, ISBN 0-08-043025-2*.
- Kanchanasevee, P., Biswas, G., Kawamura, K. and Tamura, S., "Contract-Net Based Scheduling for Holonic Manufacturing Systems", *Proceedings of SPIE ANISMI, 15-16 October 1997*, pp. 108-115.
- Koestler A., *The ghost in the machine*, Editions Arkana, Londres, 1989
- Pujo, P., Broissin, N. and Bertrand, J.C., "Simplification de la programmation de tâches dans les Systèmes Automatisés de Production Flexible, par l'intégration de primitives métier dans les machines de production", *5ème Congrès International de Génie Industriel*, pp 53-61, Grenoble, 2 - 4 avril 1996.
- Smith R.G., "The contract net protocol: high level communication and control in a distributed problem solver", *IEEE Trans. on Computers*, Vol. 29, 1980, pp 1104-1113.

Hardware in the Loop Simulator for AUS-UAV System

Hasan M M¹, and Al-Jarrah M A²
Mechatronics Center, American University of Sharjah

Abstract

This paper is reporting the creation of the HILS setup to simulate the performance of the AUS-UAV system and the ground station. The paper will give an overview of the integration of the Ground Station created using a GUI IDE, the Avionics Unit created using an embedded system, integrated with various sensors, and the Simulator created using Simulink/Flight Simulator combination. The simulation will show the ground station and the embedded system working cooperatively to guide the UAV through several mission waypoints. The paper ends with the algorithm used to create the path that the UAV will fly over.

1. Introduction

Autonomous aerial vehicles have tremendous appeal due to the fact that they can operate with relative autonomy (minimal human intervention). They have a large number of applications for them in the military, civilian, and commercial field. Some examples of these applications include surveillance, tracking an object, crop dusting, or in case of the military, provide air support. This paper is reporting the progress on the work done as part of the continuation of the AUS-UAV project [1] and the standalone Ground Station project [2]. The main focus is the creation of a working model of a Ground Station with bidirectional communication facility between itself and the UAV that allows the user to track the UAV on a map, and also direct the UAV to certain destinations by having it follow a set of waypoints (path planning).

Currently the UAV exists with the sensors, embedded system, and the actuators all integrated together to create a dynamic flight controller (DFC) capable of maintaining the stability of the UAV during flight. A lateral and longitudinal autopilot also exists as guidance controllers. Missions are currently “hard-coded” into the flight controller. This is a not an efficient mission planning capability since the UAV has to be dismantled every time a new mission has to be reprogrammed into the embedded system. Communication between the user and the UAV is facilitated by a text-based interface (TBI) through HyperTerminal, with the embedded system providing the text menu. This method of interaction is not effective as it is tedious and slow, and error prone. Since the embedded system is providing the menu, no flight control can occur during the communication process. The Standalone ground project station was a simulation only project with no actual UAV hardware implemented or, and addresses only the communication aspect between the simulated UAV and the Ground Station.

To address these shortcomings a GUI based Ground Station has been designed, and a trajectory controller with autonomous path planning ability has been integrated with the lateral/longitudinal autopilots in the flight controller. The new interface of the Ground Station

¹AUS Mechatronics Graduate Students

²AUS Mechatronics Director and Professor

will be graphical, and therefore easy and intuitive to use. The user is spared the need to know any low-level flight dynamics, and instead can spend his/her time completing high-level mission objectives such as selection of waypoints, and other mission parameters. The Ground Station will allow flexible mission planning capabilities through the dynamic path-planning feature that will be incorporated. A set of waypoints will be translated to a flight path that the UAV will follow. New missions can be entered without the need to reprogram the embedded system onboard the UAV. Building the communication interface in the Ground Station also frees up the embedded system on board the UAV, allowing it to provide continuous flight control.

2. Flight Path Planning

Path planning in a UAV provides the level of autonomy by having minimal ground control. In an abstract term, path planning involves creating a plan to guide a point-like object from its initial position to a destination waypoint. Along the way, there may be a set of regions to visit and a set of regions to avoid. In addition, the traveling object may have certain motion constraints. Path planning strategy could either be a static or dynamic depending on whether the path-planning problem is to create a path in static or dynamic environment [3].

In [4] a behavioral approach to path planning is considered by creating a set of flying modes (or behavior) that will guide the aircraft to its next waypoint. This approach works best for helicopters, since they have the hover mode. Behavior includes different flying modes such as take off, cruise, turn and landing, which can be used to compose an entire flight path.

In [5] a concurrent constraint programming (CCP) was used as the main tool for the design and the implementation of a software path planner. CCP is a very high level and complex heuristic path planner that takes into account obstacle avoidance, shortest and optimum flight path, and weighed regions. Weighed regions are regions with abnormally low or high pressure, wind speeds, or any other factor affecting flight.

Most path planning follows an approach where the path planning, trajectory smoothing, and flight stability are separated into separate layers [6, 13, 14]. The path planner merely plots a series of set points between two waypoints, navigating through obstacles, whereas the trajectory controller guides the UAV through a smooth path through those set points. The stability controller runs in the background, and keeps the aircraft stable through flight.

A final issue that needs to be addressed is the position of the aircraft during flight. The GPS unit onboard the UAV receives a position fix once every second. With a UAV top speed of 25-30 m/s, the trajectory controller does not get the current GPS coordinates fast enough to guide the UAV, therefore the UAV can stray from its path during flight. To correct this problem the UAV needs a faster and an accurate method of determining its position without the need of GPS signals. A dead reckoning navigation system based on the fusion of inexpensive inertial (INS), air data, and magnetic sensors to is described in [7]. This GPS/IMU combination is also used in [8] to create guidance, navigation, and control (GNC) algorithm.

As can be seen from above the major part of this project is to create a suitable path for the UAV to follow. This is no mean task, since it requires complex algorithm to generate paths that meet all possible constraints place on it. Several path-planning algorithms exists, among them the Dijkstra's Algorithm [9], which calculates the shortest path, by evaluating all possible paths from a point and choosing the shortest path available at that point. However Dijkstra's Algorithm does not give a global optimum, since the path look ahead is only one point. [10] describe a highly autonomous and flexible guidance system that utilizes onboard flight path prediction in combination with numerical optimization routines to guide the vehicle. For this paper, the following algorithms were researched:

- A guidance strategy based on the UAV instantaneous velocity vector, the position vector of the target from the UAV, and instantaneous turn rate. This strategy can steer the UAV from a given initial position and heading to a specified destination, in an obstacle free 2-D (constant altitude) environment, provided the algorithm is given continuous position and heading updates [1, 10]. Since our current goal is to allow the UAV to fly between waypoints without changing altitude, this algorithm may be suitable.
- In [11], a technique for planar trajectory is proposed where the trajectory is modeled as a planar spline. The goal is to have the UAV follow a reference trajectory specified by a B-cubic spline, which is said to be easy to construct using minimal user input and requires minimal computation. The spline is also developed in a constant altitude plane.
- In [12], the trajectory has been created using a "Clothoid" or "Cornu-Spiral" algorithm. Clothoids are arcs whose curvature increases with arc length. The justifications for using Clothoids were low computation time, and the fact that objects moving in a straight line (zero curvature) cannot instantaneously enter into an arc with a certain curvature.
- In [13, 14], a κ -trajectory is used to join to straight-line segments to provide a dynamically feasible path for the UAV.
- In [15], a sinusoidal path was developed for the UAV to track, if the UAV response is fast, or if the waypoint is far compared to current UAV position. This ensures the shortest possible path and therefore saves a lot in terms of fuel consumption. A surveillance algorithm in the form of a rose curve, shaped like an '8' is also described. This is useful since a plane, unlike a helicopter cannot hover over a position indefinitely.

3. HILS Simulator Setup of UAV System

Figure 1 below shows the setup of the hardware in the loop simulator that is used to test the functionality of the Ground Station, and the trajectory controller, autopilot and flight controller implemented in Avionics Unit's embedded system. This simulator models an actual UAV in flight performing a mission. Mission consists of a set of waypoints that the UAV must fly over in sequence. There are four components to the setup:

- Ground Station: This unit will display the status of the UAV to the user using various graphics and text based objects. The Ground Station will also allow the user to select waypoints and enter other mission commands. The Ground Station should allow the user to plan mission, by displaying the path taken by the UAV for a given set of waypoints.
- Avionics Unit: This unit is placed onboard the UAV and is responsible for taking various flight parameter measurements. The embedded system in it is also responsible for maintaining flight stability, and guiding the UAV on its mission.
- Simulink Model: This unit contains the mathematical model of all the sensors, actuators, and engines onboard the UAV. Simulink also contains the atmospheric, earth, and flight dynamic models. Basically the Simulink model will simulate the actual UAV in flight, and will respond based on the actuator commands (PWM signals) from the Avionics Unit [1].
- Flight Simulator: This unit is there to show the user a visual representation of the UAV in flight [2].

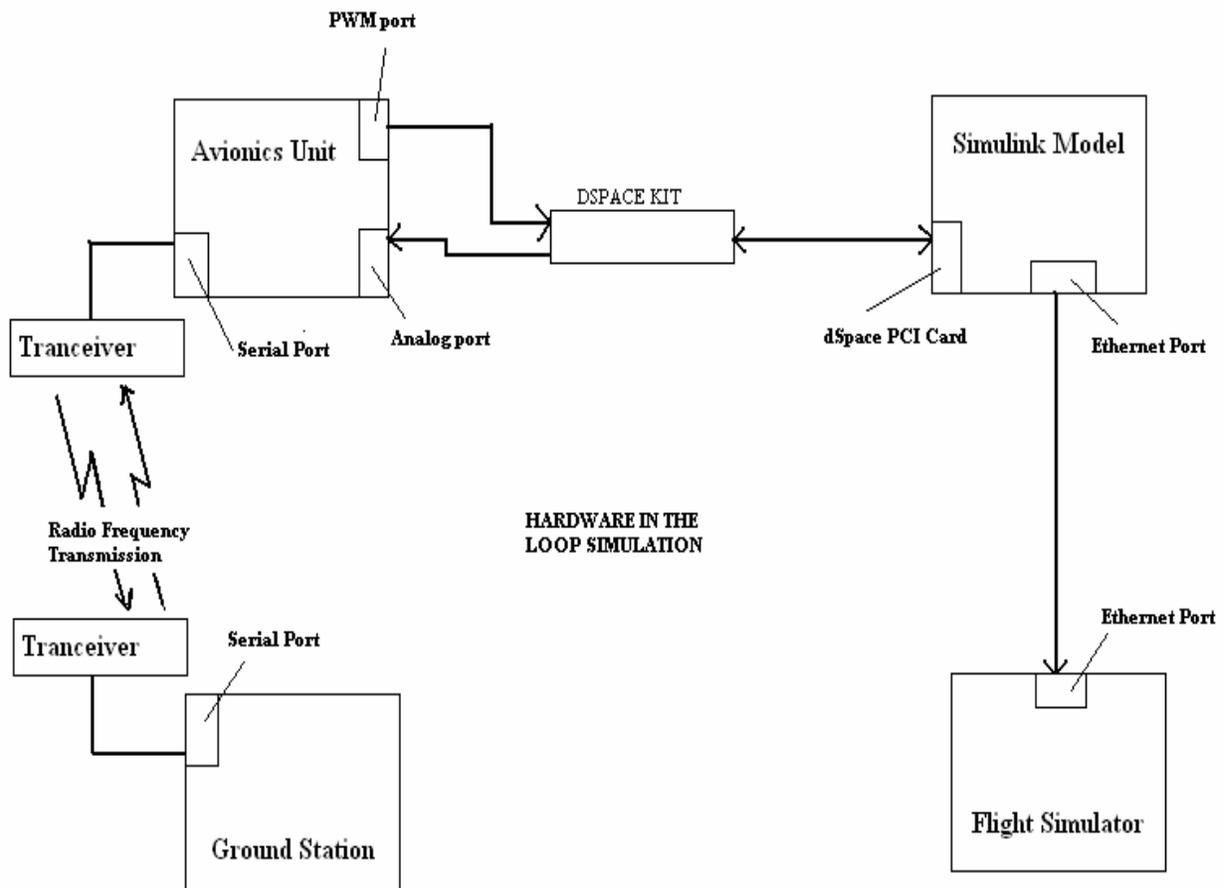


Figure 1: The HILS setup of the Ground Station and Avionics Unit

The user will select waypoints in the Ground Station, generate a path, verify that the path is suitable, and then start the mission, which will be the start of the simulation. During the simulation, the Ground Station will communicate the waypoints to the embedded system in the Avionics Unit via an RF transceiver. The trajectory controller will utilize these waypoints to guide the 'UAV' that is modeled by Simulink.

The Simulink model will communicate the simulated sensor data to the Avionics Unit via the digital to analog converter of the dSpace kit. The flight controller programmed in the embedded system will utilize these data to generate control signals to stabilize the aircraft. The control signals are servomotor commands generated using Pulse Width Modulation. These PWM signals are input to Simulink via the digital input port of the dSpace kit. The Simulink will run these PWM signals through its servomotor model, and then simulate the motion, position, heading, and orientation of the UAV. Heading (Compass) and position (GPS) information are input back to the embedded system via the serial port in the dSpace kit using RS-232.

The trajectory controller programmed in the embedded system will use the heading and GPS location data to guide the UAV to its next waypoint. In the same time the embedded system in the Avionics Unit will communicate the sensor data to the Ground Station via RF transceiver. The Ground Station will utilize these data to display the position of the UAV on the map, the heading and orientation in the compass and artificial horizon respectively. The user can verify if the UAV tracks the generated path on the map, and can utilize this information to tune the path algorithm. The flight simulator will show visually the orientation of the UAV. The orientation data will be transmitted to the flight simulator program by Simulink via the Ethernet port.

4. Ground Station Design

Ground stations are very important to monitor the state of flying objects such as satellites, missiles, and aeroplanes to name a few. For an UAV, the Ground Station provides important information such as pitch, roll, yaw, angle of slip, and angle of attack. The Ground Station also provides information regarding heading, location, altitude, and velocity. Very often, it is not possible to track an UAV in flight visually if it is far away; therefore, a Ground Station can be both valuable for the user and the UAV. A passive Ground Station merely monitors the UAV, but an active Ground Station should also be able to direct and guide the UAV through its mission, such as surveillance, path planning, calibration, and other functions.

Figure 2 below shows the layout of the Ground Station designed using Visual C++ 6.0. This is a graphical user interface that provides the user with easy to understand visual data about the status of the UAV. The map on the top center of the display shows the location of the UAV on the map of a given area. The flight instrument panels to the lower right and left of the screen show the UAV airspeed, altitude, heading, and orientation respectively. The textual display in the lower middle portion displays other information such as battery power level, fuel level and other information, which will not be a part of this simulation.

The most important part of the Ground Station is the ability to plan missions. The user can select the waypoints in the map, which will be converted to GPS coordinates, and then transmitted to the Ground Station via the serial interface. Once the waypoints (a maximum of five for the purpose of this simulation) are selected, the user can have the Ground Station generate the approximate path the UAV will take during flight. During the actual mission the user can ascertain how well the UAV tracks the given path.

Note that the Ground Station will only communicate the waypoints to the Avionics Unit, not the entire path. The Avionics Unit will run a similar algorithm as the Ground Station to guide the UAV to the next waypoint. The path generated by the Ground Station is constrained by the physical limitations of the UAV. Therefore the user is not shown any path that will be impossible for the UAV to track. For this simulation the path will consist of straight lines and circles, which the UAV can execute properly. There will be no obstacles for the UAV to navigate through.

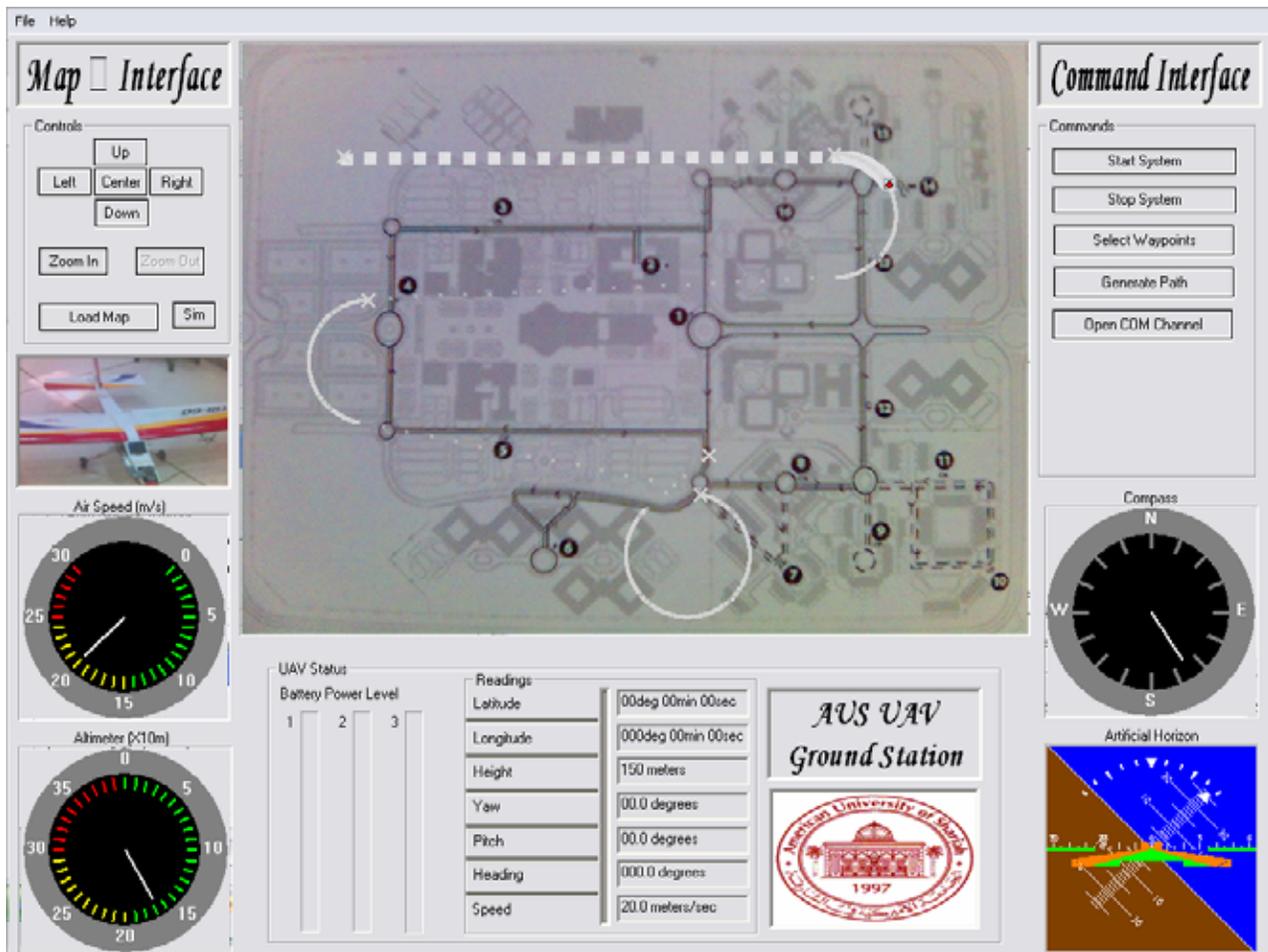


Figure 2: Framework of the Ground Station GUI display

5. Avionics Unit

The Avionics Unit consists of various sensors, and actuators integrated together with the embedded system. The embedded system consists of two 16-bit Motorola 68HC12 microcontrollers running at 24 MHz. The sensors used are three (x, y, z) rate gyros to provide rate of turn (such as yaw) of UAV, three (x, y, z) accelerometer to provide information about the forces acting on the UAV, absolute pressure sensor to provide altitude of UAV, gauge pressure sensor to provide air velocity of UAV, tilt sensor to provide pitch and roll angle of UAV, angle of attack sensor, compass providing heading of UAV, and GPS providing location of UAV. The flight controller program in the embedded system uses this information to keep the UAV stable by generating PWM signals to actuate the servomotor.

There are also two autopilots implemented in the embedded system: the lateral and longitudinal autopilot. The longitudinal autopilot stabilizes the UAV at a given altitude, or controls the rate of climb of the UAV. The longitudinal autopilot on the other hand stabilizes the UAV in a given heading or controls the rate of turn of the UAV. These autopilots run in parallel to achieve a three dimensional maneuver of the UAV. For the purpose of the simulation the altitude will be kept constant, while the UAV completes a mission.

Therefore the trajectory controller implemented in the embedded system will utilize the lateral autopilot to have the UAV either move in a straight line or turn in a circular path depending on the best method to reach the next waypoint. For this simulation, the sensor bank will be dismantled from the embedded system, and instead Simulink will provide the simulated sensor information.

6. Simulink and Flight Simulator

The Simulink model is implemented using the Aerosim Blockset version 2.8. Figure 3 shows the implementation below. The model takes as input the PWM signals as servomotor command. The model outputs the various sensor data that are simulated, including analog data from gyros, accelerometers, tilt sensors, compass, and GPS sensor

These sensor values are simulated using aerodynamic the model of the TRI-60 UAV, the propulsion model of the engine, the atmospheric and earth model, and a model that solves the equations of motion. The servo inputs signals are PWM signals generated by the embedded system of the Avionics Unit. Simulink simulates the behavior of the UAV given the servo-commands. Simulink outputs this information back to the embedded system in the Avionics Unit as sensor information. Gyro, accelerometer, tilt, speed, and height information are output as analog signals, while the heading and position information is output serially as RS-232. This ensures that the outputs conform to the actual sensor outputs.

Simulink also output the speed, tilt, position, and heading information via the Ethernet port to flight simulator program. Based on this information, the flight simulator visually represents the UAV orientation in flight.

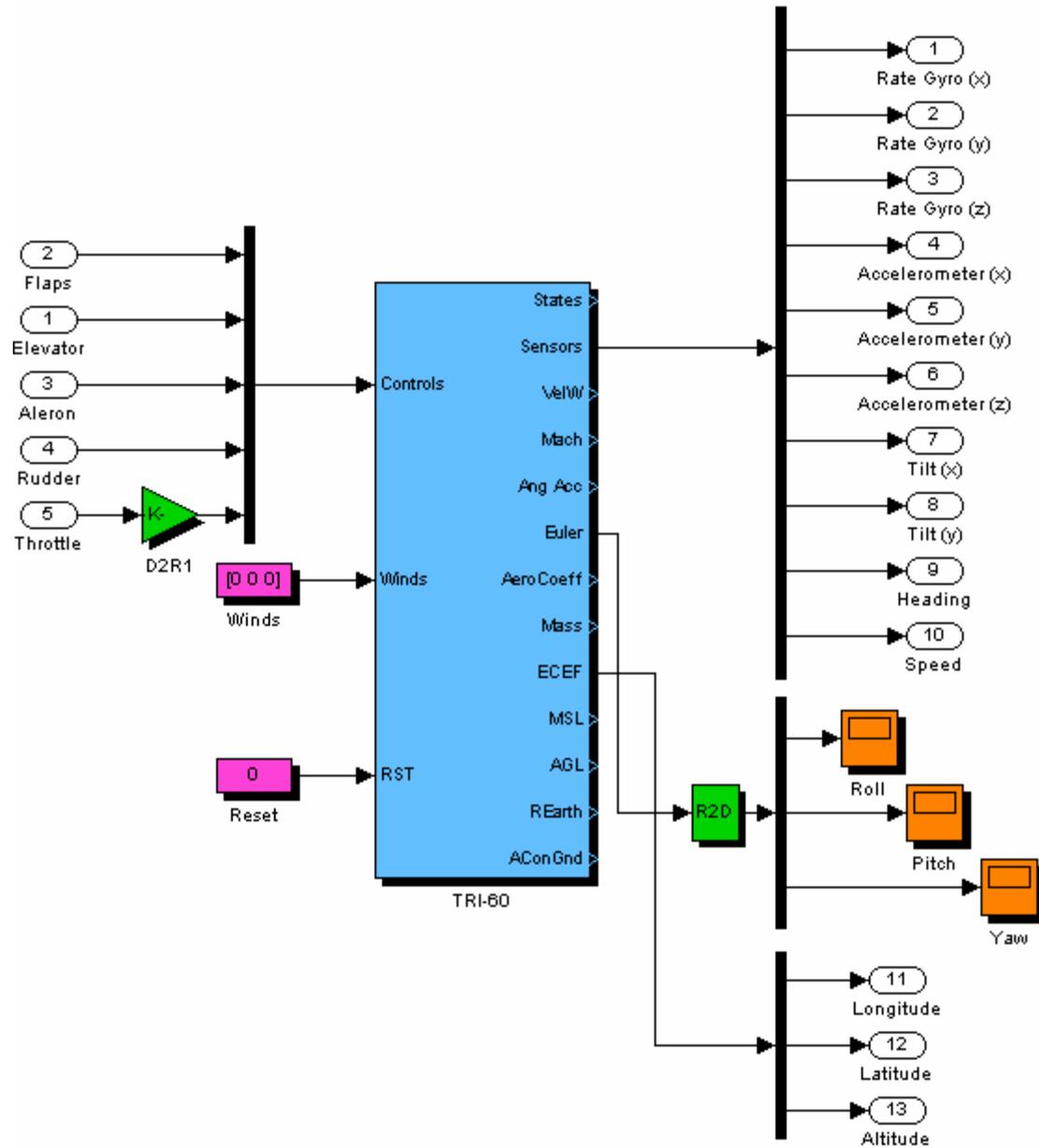


Figure 3: The Simulink model of the TRI-60 UAV implemented using Aerosim Blockset

7. Path Algorithm and Constraints

The path-planning algorithm must take into account the physical constraints of the UAV. Among these are: the top speed, the minimum flying radius, the stall speed, and the maximum bank angle to name a few. Equation 1 shows the minimum radius R , at which the UAV can fly at a given speed V , and bank angle ϕ .

$$R = \frac{V^2}{g * \tan \phi} \quad (1)$$

Equation 2 shows the minimum speed at which the UAV can fly when banked at an angle ϕ .

$$V_{STALL} = \sqrt{\frac{2 * W}{\rho * C_L * S * \cos \phi}} \quad (2)$$

Where W = total weight of the UAV, ρ = air density, C_L = lift factor, S = surface area of the wings.

The other constraint is that the UAV must fly at least 1.25 times its stall speed for safety reasons, and also the UAV should fly at 80% of its maximum speed to prevent the engine from overheating.

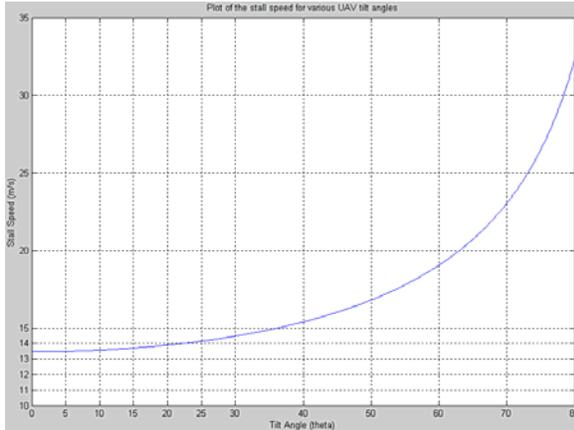


Figure 4: Plot of stall speed vs. bank angle

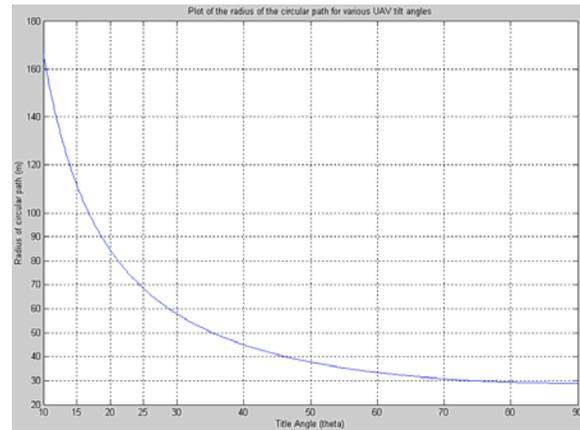


Figure 5: Plot of flying radii vs. bank angle

Figure 4 and 5 above shows the plots of stall speed and flying radius at various bank angles for our UAV. Since the maximum speed of the UAV is 25m/s, a speed of 20m/s is maintained during flight. At that speed, the maximum bank angle is 40 degrees, taking into account that the UAV is flying 1.25 times the stall speed (16m/s). Using that speed the minimum-flying radius is 45m. The path algorithm must ensure that a path do not have a radius of less than 45 meters.

Using the constraints above, the kinematic model of the aircraft can be defined as:

$$\begin{aligned} \dot{x} &= V \cos \phi \\ \dot{y} &= V \sin \phi \\ \dot{\phi} &= u \end{aligned} \quad (3)$$

Where x and y are the change in position given a turn rate of u . U must be smaller or equal to the maximum turn rate M , which is given by:

$$M = \frac{g \sqrt{n^2 - 1}}{V}, \quad n = \frac{L}{W} \quad (4)$$

Where n = load factor of the UAV, L = total lift, and W = weight of UAV.

The path algorithm followed here is three-step process. It consists of straight lines and circular curves in each leg of the UAV mission. The UAV will approach a waypoint following a straight line, upon which it will execute one of the three maneuvers:

- If the next waypoint is within ± 20 degrees of the UAV current heading, it will bank slightly to the required heading and follow a straight line to the next waypoint.
- If the next waypoint is more than 20 degrees of the UAV current heading, the UAV will fly a circular curve until it faces the next waypoint, and then fly straight to the waypoint.

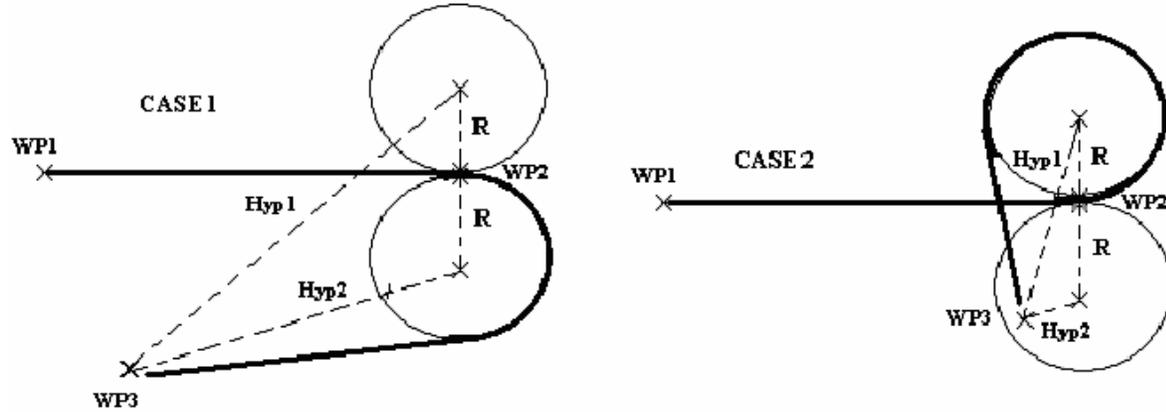


Figure 6: Path taken by UAV when entering a curve.

Figure 6 above shows the decision taken by the path algorithm when turning. This decision is based on the distance from the center of the circular arc to the next waypoint. The center of the circular arc is given by:

$$C_{POS} = W_{POS} + R * \hat{r} \quad (5)$$

Where W_{pos} is the position of the waypoint where the turn is taken (waypoint 2), C_{pos} is the position of the center of the circular arc, and \hat{r} is the direction vector of the center of the circular arc from W_{pos} . If the direction vector of the current path is given by $a\hat{i} + b\hat{j}$ then:

$$\hat{r}_1 = b\hat{i} - a\hat{j} \text{ or } \hat{r}_2 = -b\hat{i} + a\hat{j} \quad (6)$$

The distance between the centers of the two circular arcs and the next waypoint (waypoint3) is found (Hyp1 and Hyp2). As can be seen in Case 1, the UAV will fly over the circular arc whose center is nearer to the next waypoint. On the other hand, if the distance between the center of the circular arc and the next waypoint is smaller than R (Case 2), then the UAV will fly over the circular arc whose center is farther from the waypoint. Whichever way the UAV turns, the direction of turn (clockwise or counterclockwise) depends on the cross product of the direction vector of the current path and \hat{r} . A negative cross product signifies a clockwise turn, whereas a positive cross product signifies a counterclockwise turn.

Once the UAV is in a turn, it will iteratively compute the heading from its current position in the arc to the next waypoint, and compare this heading with its own current heading. The heading between the current position in the arc and the next waypoint is computed by:

$$H = \tan^{-1}\left(\frac{y_p - y_c}{x_p - x_c}\right) + 45 \quad (7)$$

Where X_p , Y_p are the coordinates of the next waypoint and X_c , Y_c are the instantaneous coordinates while flying over the circular arc. The heading is computed from North in a clockwise direction. The current heading of the UAV is taken from the compass. If the two headings are within ± 1 degree, then the turn is completed and the UAV flies straight to the next waypoint. This algorithm, although providing a minimum flying path, has the constraint that the UAV heading when it reaches the next waypoint cannot be controlled.

8. Conclusions and Future Work

The setup presented in this paper is merely a simulation that tests the functionality of the actual hardware including the Ground Station and the embedded system in Avionics Unit. The work described in this paper is part of a larger project with the aim of flying the UAV autonomously through several mission waypoints and monitoring its progress in the Ground Station. The path algorithm given in this paper bring the UAV to the next waypoint while flying over minimum distance, but the heading of the UAV when it reaches the next waypoint cannot be controlled. There may exist situation where it is imperative not only to fly the UAV over a certain waypoint, but also to fly pointing towards a given heading. Another area where future contributions can take place is to recreate this algorithm to fly the UAV in a three-dimensional space, since currently the altitude is kept constant.

Reference:

- [1] Hadi, M. "Autopilot Design Based on Fuzzy Supervisory Controller for AUS-Autonomous Vehicle", Master Thesis, Spring 2005, American University Sharjah
- [2] Wael, S., Mahmoud, G. "Implementation of Ground Station to Control Simulated UAV", Senior Design Project, Fall 2004, American University of Sharjah
- [3] Bhat, S. P., Kumar, P. "A Feedback Guidance Strategy for an Autonomous Mini-Air Vehicle". National Conference on Control and Dynamical Systems, January 27-18, 2005, IIT Bombay.
- [4] Egerstedt, M., Koo, T. J., Hoffmann, F., Sastry, S. "An Integrated Approach to Path Planning and Flight Controller Scheduling for an Autonomous Helicopter". Proceedings of the 7th Mediterranean Conference on Control and Automation, August 1999, Haifa, Israel
- [5] Stefano G., Tranchero, B. "Concurrent constraint programming based path planning for uninhabited air vehicles". SPIE/Defense and Security Symposium on Unmanned Ground, Ocean, and Air Technologies, April 2004, pg 12 – 16
- [6] Rem, W., Beard, R. W. "Constrained Nonlinear Tracking Control for Small Fixed wing Unmanned Air Vehicles". American Control Conference, 2004, Boston, MA, pg: 4663 – 4668.
- [7] Egziabher, D. G., Power, J. D., Enge, P. K. "Design and Performance Analysis of a Low Cost Aided Dead Reckoning Navigation System". Gyroscopy and Navigation, 2001, Volume 4, No: 35, pg: 83 - 92

- [8] Kim, J. H., Wishart, S., Sukkarieh, S. "Real Time Navigation, Guidance, and Control of an UAV using Low cost Sensors." International Conference of Field and Service Robotics (FSR03), July 2003, Yamanashi, Japan, pg: 95 – 100.
- [9] Barbehenn M. "A Note on the Complexity of Dijkstra's Algorithm for Graph with Weighted Vertices". IEEE Transactions on Computers, February 1998, Volume 47, Issue 2, pg 263.
- [10] Graesslin, M., Schoettle, U. "A NLP based reentry flight guidance algorithm", 18th International Symposium on Space Flight Dynamics, October 2004, Munich Germany
- [11] Harbick, K., Montgomery, J. F., Sukhatme, G. S. "Planar Spline Trajectory Following for an Autonomous Helicopter". Journal of Advanced Computational Intelligence – Computational Intelligence in Robotics and Automation, May 2004, Volume 8, Issue 3, pg: 237 – 242
- [12] Chean S. L., Doria, M., Chan, J., Parker S., Goatley, S. "Path Planning and High Level Control of an Unmanned Aerial Vehicle". Ph.D. Thesis, Fall 2002, Spring University of Sidney"
- [13] Kingston, D. B. "Implementation Issue of Real-Time Trajectory Generation of Small UAV". Master Thesis, 2004, Birmingham Young University
- [14] Beard R., Kingston, D., Quigley, M., Snyder, D., Christiansen, R., Johnson, W., McLain, T., Goodrich, M. "Autonomous Vehicle Technologies for Small Fixed Wing UAV". AIAA Journal of Aerospace Computing, Information, and Communication (to appear)
- [15] Lee, J., Huang, R., Vaughn, A., Xiao, X., Hedrick, J. K. "Strategies of Path Planning for a UAV to Track a Ground Vehicle". 2nd Annual Symposium on Autonomous Intelligent Networks and Systems, June/July 2003, Bologna, Italy

Free forming of microstructures using synchronous localized electro-deposition

R.A. Said

United Arab Emirates University,
Department of Electrical Engineering
P.O. Box: 17555, Al-Ain, UAE
rasaid@uaeu.ac.ae

Abstract

To further advance the capabilities of the localized electro-deposition (LED) technology for the fabrication of micro-structures, this work presents the utilization of a synchronized linear displacement of the deposition electrode to allow the fabrication of axially-symmetric high-aspect-ratio micro-structures with controllable cross sectional geometries and dimensions. The proposed synchronized electrode positioning was implemented using continuous and stepping displacement algorithms. The continuous electrode displacement algorithm allows deposition at all points along the structure boundary while the electrode is in motion. The deposition current is continually monitored while the electrode lateral displacement is maintained at a specified speed along the geometrical trajectory. Electrode withdrawal in the vertical direction is triggered upon sufficient deposition at all points along the trajectory. In the stepping displacement algorithm, however, the electrode is moved from a point on the lateral trajectory to a neighboring location only upon sufficient deposition at the current location. Once enough deposition occurs at all points along the trajectory, the electrode is withdrawn in the vertical direction. Deposition Comparison between both algorithms is extracted from SEM images in terms of deposit characteristics and required deposition time.

1 Introduction

Although was introduced only a decade ago (Hunter, I.W., 1997) localized electro-deposition (LED) has a great potential to provide solutions to the various challenges (Smith, H.I., 1990 and Havemann, R.H., 2001) facing currently adopted fabrication technologies (Van Dyke, L.S., 1992, Reynarts, D., 1997, Dizon, R., 1993, Sun, X.-O., 1996, Wallenberger, F.T., 1994, Madiu, M.J., 1997, Ikuta, K., 1998, and Kikuchi, T., 2003). The ability of LED to localize an electrolysis reaction (El-Giar, E.M., 1997), that results in material deposition at a miniature spot determined by the size of an electrode-tip, provides great hopes for controlled fabrication at the sub-micrometer and nanometer scales. Utilizing micromachined probes with ultra sharp tips (Madden, J.D., 1996), such as those used in various types of scanning probe microscopy, micro-fabrication by LED would overcome fundamental limits facing lithography-based and pattern transfer techniques (Rai-Choudhury, P., 1997 and Timp, G., 1999). Moreover, the tip directed deposition in LED allows selective growth of structures, an issue that is still a serious barrier for the utilization of the newly

developed and promising techniques of nanotubes and nanowires (Wildoer, J.W.G., 1998 and Odom, T.W., 1998).

In addition to the technical and economical advantages that are characteristics to the electroplating technology (Durney, L.J., 1984) LED as a mask less process (Bard, A.J., 1990) can contribute to achieve shorter development times with a better economical return when considering post processing and rapid prototyping applications (Dean, R.N., 2000). Achievable deposition rates are several $\mu\text{m/s}$ and can reach greater than 30 $\mu\text{m/s}$ with the aid of a jet mechanism (Gelchniski, M.H., 1985). Although most demonstration of LED involved the deposition of metallic structures, LED is applicable to various materials without limitations on the type, size, or thickness of both the substrate or the deposited material (Paunovic, M., 1998). Above all, one of the most important features of LED is its ability to produce high aspect ratio structures (El-Giar, E.M., 2000).

The first demonstration of the LED technique (Madden, J.D., 1996) has attracted attention and became the focus of several efforts that contributed to its development. Some of the invested development efforts focused on investigating the effect of different parameters of the LED process on the deposition rate and characteristics of deposited structures. Among the investigated parameters are the applied electrode potential and electrolyte concentration (El-Giar, E.M., 2000), the presence of organic additives in the electrolyte (Jansson, A., 2000), the effect of insulation material surrounding the electrode tip (Said, R.A., 2001, II), the use of non-conducting substrates (Heb, C., 1997), and the utilization of glass micropipettes in replacement of the conducting tip (Muller, A.D., 2000). Other efforts have investigated the utilization of mechanical methods, to enhance deposit characteristics and resolution, including: the utilization of rotor electrode (Yeo, S.H., 2001) which creates a uniform deposition field to overcome defects in the electrode tip, and the use of ultrasound vibrations during deposition (Yeo, S.H., 2002) to enable the regular removal of air bubbles which result during deposition and often block deposition growth. Further development efforts have focused on providing an understanding of the deposition mechanism by providing a process model and quantitative evaluation of the deposition profile (Said, R.A., 2001, I, 2003).

All the above indicated efforts have thus far contributed to three main aspects of the LED: process description, process parameters optimization, and process modeling. Recently, a remarkable advancement of the technology in terms of process repeatability has been presented (Said, R.A., 2003). Process automation has also been addressed (Said, R.A., 2004). A crucial issue, however, still remains to be resolved before LED is advanced to become a standard fabrication process that can be integrated with current commercial fabrication technologies. This issue is identified as the degree of complexity of enabled geometries. Demonstrations of LED thus far involved mainly one dimensional (1-D) structures such as columns and helices, which are suitable for limited applications (Rebeiz, G.M., 1992). Several micro-systems, however, require two dimensional (2-D) microstructures (Gebhard, M., 1995, Van Der Schoot, B., 2001, Ikuta, K., 1994, and Roberston, J.K., 1994), while advanced applications require even three dimensional (3-D) structures described by curved surfaces.

The limited 1-D capabilities of LED demonstrated thus far is a result of the algorithm used to control the positioning of the electrode-tip as well as the positioning system utilized in the experimental setups. In these systems, the electrode-tip is withdrawn away from the deposition end upon sufficient deposition (indicated by a sudden increase in the deposition current) beneath the tip. Tip withdrawal is activated in the vertical direction only, thus resulting in 1-D column like structures.

To enable the deposition of 2-D and 3-D structures, the deposition should be activated over a trajectory or a selected area of the substrate surface. Tip withdrawal in these cases can be triggered only upon sufficient deposition along the entire trajectory or the selected area of the substrate. This requires positioning system with at least three degrees of freedom with a closed loop feedback control system to maintain accurate positioning of the tip. Moreover, a control algorithm that monitors deposition during the entire travel of the tip is required.

This work proposes the utilization of a synchronized linear displacement of the deposition electrode to allow the fabrication of micro-structures with controllable cross sectional geometries and dimensions. For this purpose, an advanced positioning system with an optical encoder based feedback system was used to provide absolute positioning with nanometer accuracy. Two positioning control algorithms were used; continuous and stepping displacement algorithms. Comparison between both algorithms is provided in terms of deposit characteristics (extracted from SEM images) and required deposition time.

2 Fabrication by LED

The schematic drawing in Fig. 1 illustrates a general arrangement of a typical setup used for LED. A microelectrode is placed very close to a conducting substrate while both immersed in an ionically conducting electrolyte that contains ions of the material to be deposited. The microelectrode is usually insulated from all sides except for an exposed tip region with micrometer scale dimensions. An electric potential is applied between the microelectrode and the substrate, thus causing a faradic current to flow through the electrolyte between the microelectrode and the substrate. Since the used electrolyte contains reducible metal ions (e.g., Cu^{2+} ions as in the presented work) and the substrate is connected to a negative potential with respect to the microelectrode, then the flow of faradic current results in an oxidation process at the microelectrode tip and a deposition of metal ions at the substrate (Bard, A.J., 1980), as illustrated by the circled drawings in Fig. 1.

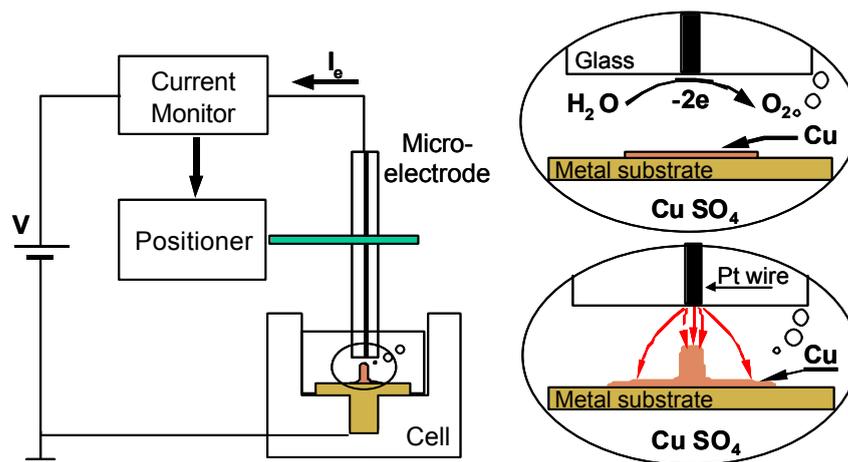


Fig. 1. An illustration of a typical LED arrangement. The concept of localized deposition is illustrated by circled drawings demonstrating oxidation reaction at the micro-electrode end (anode), thus causing bubbles, and reduction reaction at the substrate (cathode) resulting in copper deposition. Intense electric field causes faster deposition rates beneath the micro-electrode end, hence localizing the deposition.

Unlike typical electroplating methods where deposition occurs at a uniform rate on all exposed regions of the substrate, the deposition process outlined in Fig. 1 is very localized to the region beneath the electrode tip (Said, R.A., 2001, II). This is due to the highly localized electric field in the spacing between the microelectrode tip and the substrate region directly below the tip. The deposition rate in LED process is proportional to the magnitude of the deposition current which is monitored during the entire fabrication session. As the deposit grows and fills the spacing between the tip and substrate, the deposition current increases due to reduced electrolyte resistance (Yeo, S.H., 2000). Upon sufficient deposition, usually indicated by a sudden increase in the current caused by a direct contact between the tip and growing deposition-end, the electrode is withdrawn from the deposit end but along the trajectory of desired geometry. Deposition current monitoring and electrode withdrawal are usually performed using conventional analog feedback control (Said, R.A., 2001, III), as illustrated in Fig. 1.

3 Synchronized Tip Positioning

Previous demonstrations of LED utilized tip withdrawal in the vertical direction only, thus resulting in the fabrication of column like structures as shown in Fig. 2. The fabrication of more dynamic structures would require the control of the tip position in other directions, lateral X & Y directions for example.

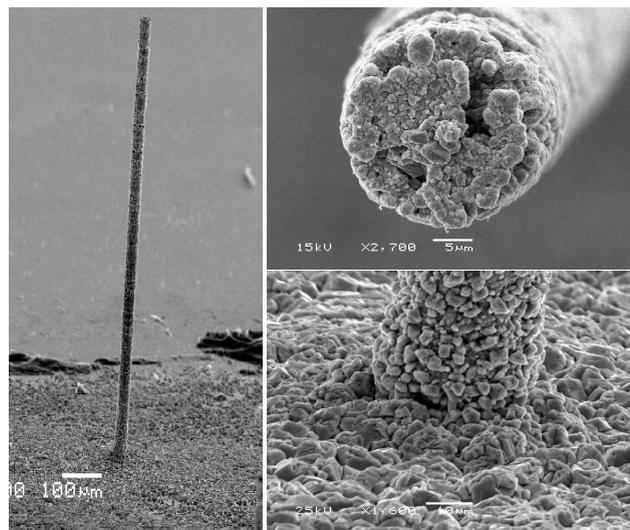


Fig. 2: SEM image of a copper column (20 μm in diameter and height of 1.2mm) fabricated by conventional LED technique that utilizes tip withdrawal only in the vertical direction.

To control the location of the tip in 3 dimensions, the positioning system must be capable of providing positioning in the lateral as well as the vertical directions, with a resolution much smaller than the tip diameter. This requires a positioning system that utilizes a closed loop feedback control system to maintain accurate positioning of the tip. Moreover, a control algorithm should be developed to monitor the deposition rate during the entire travel of the tip and synchronize the displacement of the tip. The tip synchronized displacement can be performed in two modes; stepping and continuous displacement algorithms. These are discussed next.

3.1 Tip Stepping Displacement

In the stepping displacement mode, micro-structures are treated as a stack of lateral trajectories, where each trajectory formed by deposited dots side by side, as shown in Fig. 3. Here, the tip is kept at the same location until deposition fills the spacing between the tip and substrate. Upon sufficient deposition, the positioner is triggered to place the tip at the next location in the desired trajectory. Again, the tip is kept at the same location till sufficient deposition grows beneath the tip, and then the tip is displaced at the next location. This is repeated until sufficient deposition all points along the trajectory is achieved, as shown in Fig. 3.1-3.2. Upon sufficient deposition at all points along the desired trajectory, the tip is then displaced in the vertical direction as in Fig. 3.3. Deposition along the trajectory at the new tip height is pursued as was discussed earlier in Fig. 3.1-3.2. This is repeated until desired structure height is deposited.

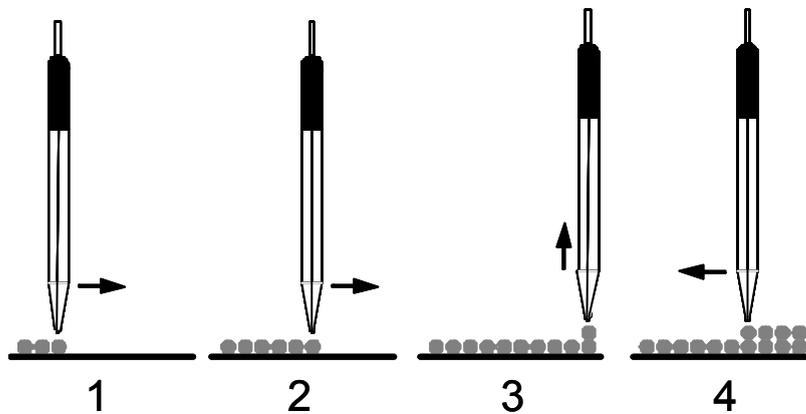


Fig. 3: A schematic representation of the tip stepping displacement mode.

The parameters of consideration in this mode of deposition that can influence the characteristics of deposition include the size of tip stepping along the trajectory, which should be within the tip diameter to assure structure continuity along the trajectory. Another factor of importance is the initial as well as the incremental tip displacement in the vertical direction, which can influence the localization of deposition as well as the speed of deposition.

3.2 Tip Continuous Displacement

In the continuous displacement mode, micro-structures are deposited while the tip is kept in continuous motion along the desired deposition trajectory, as shown in Fig. 4. The deposition is monitored during tip continuous movement, which is then displaced in the vertical direction only upon sufficient deposition is achieved along the entire trajectory, as shown in Fig. 3.1-3.4. This is repeated until desired structure height is deposited.

The parameters of consideration in the continuous displacement mode of deposition that can influence the characteristics of deposition include the displacement speed of tip along the trajectory. Similar to the stepping displacement mode, the incremental tip displacement in the vertical direction would play a major role in determining the localization of deposition as well as the speed of deposition.

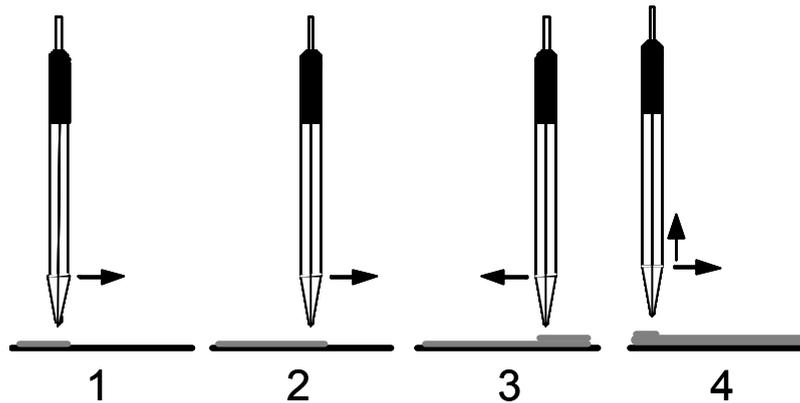


Fig. 4: A schematic representation of the tip stepping displacement mode.

4 Experimental Setup

The photographs in Fig. 5 show the details of a constructed LECD apparatus used for the fabrication of micro-structures. The apparatus consists of an aluminum base holding a deposition cell, machined from Teflon, and a three-axis positioning stage that holds the microelectrode. The three-axis positioning stage uses micro-stepping motors (New Focus, Inc., CA, USA) with optical capable encoders as feedback systems capable of repeatable incremental steps of 63 nm, achievable at various speeds with a maximum of 28 $\mu\text{m/s}$.

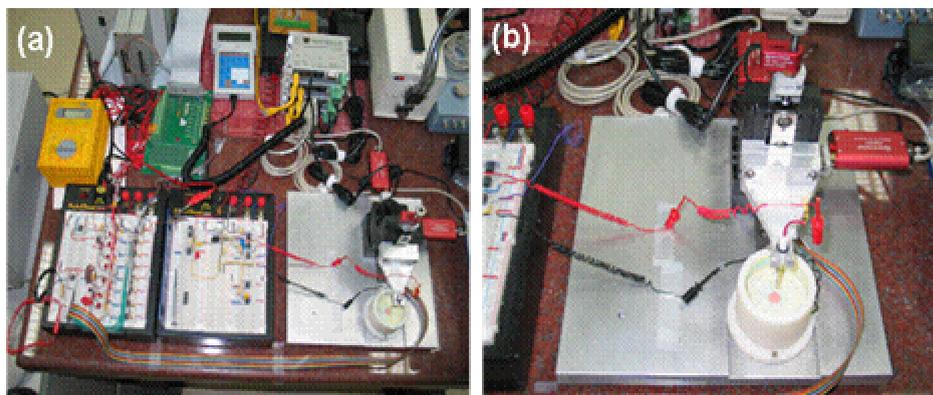


Fig. 5: Photograph of: (a) the assembled LECD setup showing the deposition stage (bottom-right), positioner control modules, and control electronics, and (b) a close-up of the deposition stage showing the deposition cell, the electrode and holder, and the three axis positioners.

The microelectrode used in the presented work was prepared from a 25 μm in diameter Pt wire following the procedures outlined in the literature. The Pt wire was inserted in a glass micropipette that was sealed to insulate the wire from all sides except at the very end of the wire, which was polished to form an exposed disk. Electrode polishing was done using successive grades of 600, 1200, and 1500 silicon carbide paper and final polishing with 1.0 μm diamond polish followed by 0.05 μm alumina to produce a smooth surface with a nanometer scale roughness. The glass surrounding the electrode wire extends to a diameter of about 1500 μm .

During deposition, an electrolyte of copper sulfate solution ($\text{CuSO}_4 \cdot 5\text{H}_2\text{O}$, 250 g/L) was used. The substrates were prepared from 1.0 cm in diameter metal rods that were cut and machined as a T-section that fits and locks into the Teflon cell. The substrates surfaces were polished with successive grades of 240, 400, and 600 silicon carbide paper. Final polishing was done using diamond polishing paste and then alumina as done in the electrode preparation.

To control the displacement of the electrode while monitoring the deposition current, a LabVIEW program was developed with an interactive interface as shown in Fig. 6. The program allows a dynamic selection of micro-structures including elliptical, rectangular, triangular, and hexagonal trajectories. The two modes of electrode displacement are available with the displacement speed of the continuous mode can be determined. Upon fabrication, the deposition cell is filled with the electrolyte and the electrode tip brought close to the substrate surface. An electric potential is then applied to the tip with the substrate grounded and a data acquisition board is used to extract the deposition current and used for feedback purposes. The feedback operates by setting a reference potential V_0 (equivalent to a reference deposition current I_0) and comparing it with the monitored deposition current. As deposition is started, copper grows on the substrate beneath the tip, causing the deposition current through the tip I_t to increase till the growing copper touches the tip at which I_t exceeds I_0 . This triggers the feedback circuit to move the tip away from the deposition end and along the desired trajectory according to the selected displacement mode. The interface has a graphical capability of showing the current location of the tip, as well as time charts of the acquired deposition current signal and the control signals sent to the three axis positioners.

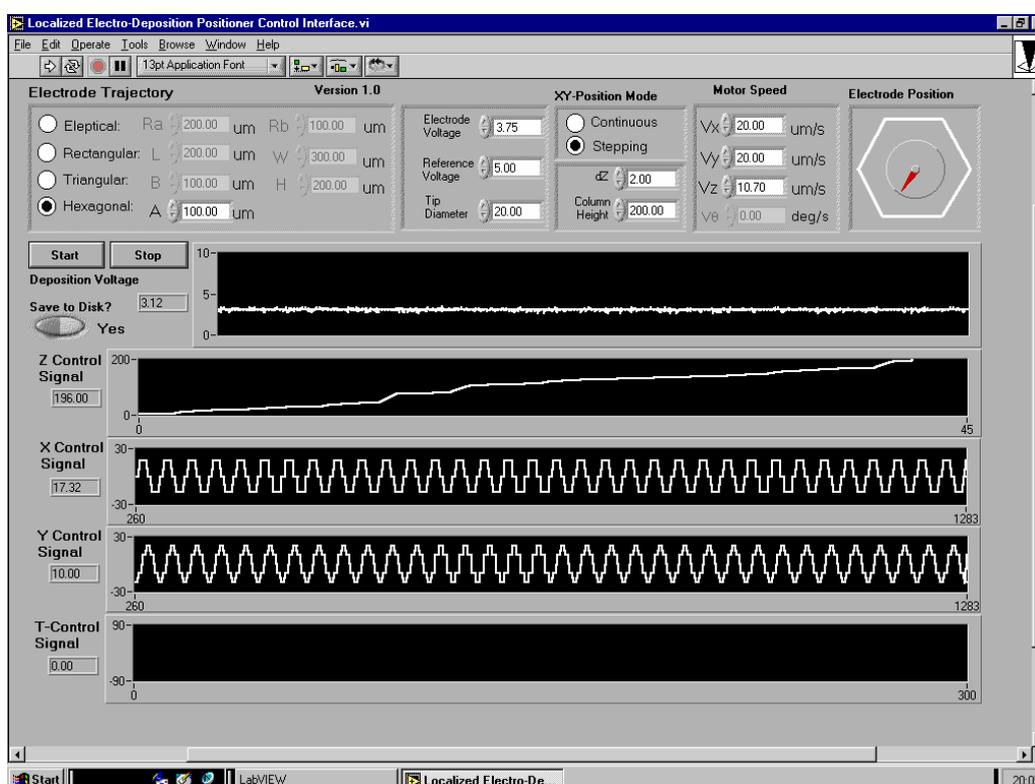


Fig. 6: A snap shot of the interface of the LabVIEW interactive program used to control the electrode displacement while monitoring the deposition current.

5 Experimental Results

To demonstrate the synchronous displacement method, deposition of micro-structures with various geometries were attempted using both the stepping and continuous displacement techniques. In all deposition attempts the electric potential between the tip and the substrate was set to 3.75 V, while the reference potential V_0 was set to 5 V. The size of the incremental stepping in the vertical direction was set to 1 μm to improve the localization of deposition and increase the deposition rate.

First deposition of structures using the stepping displacement mode was attempted with a step size of 25 μm (equivalent to the tip diameter), while selecting a rectangular geometry that has a length of 400 μm and a width of 200 μm . Figure 7(a) shows the deposition result with the structure height reaching 100 μm . The results clearly demonstrate the stepping effect as dots forming the structure boundaries. This can be improved by selecting a step size smaller than the tip diameter. Other geometries were attempted also including elliptical, triangular, and hexagonal, as shown in Fig.'s 7(b), (c), and (d), respectively.

Deposition using the continuous displacement mode was also attempted with deposition settings similar to that used for the stepping mode. Specific to the displacement mode, the displacement speed was set to 20 μm , which is the maximum travel speed attainable by the positioners. This extreme value was used for comparison purposes with the stepping mode, which can be simulated by the continuous mode by setting the travel speed to zero. The deposition results are shown in Fig. 8(a) and 8(b) for elliptical and rectangular geometries, respectively. The results in Fig. 8 suggest that the stepping effect disappears when using the continuous displacement mode. However, the deposition growth in the vertical direction is extremely slow. This is explained by the selected fast travel speed of the tip, and thus can be improved by reducing the tip travel speed.

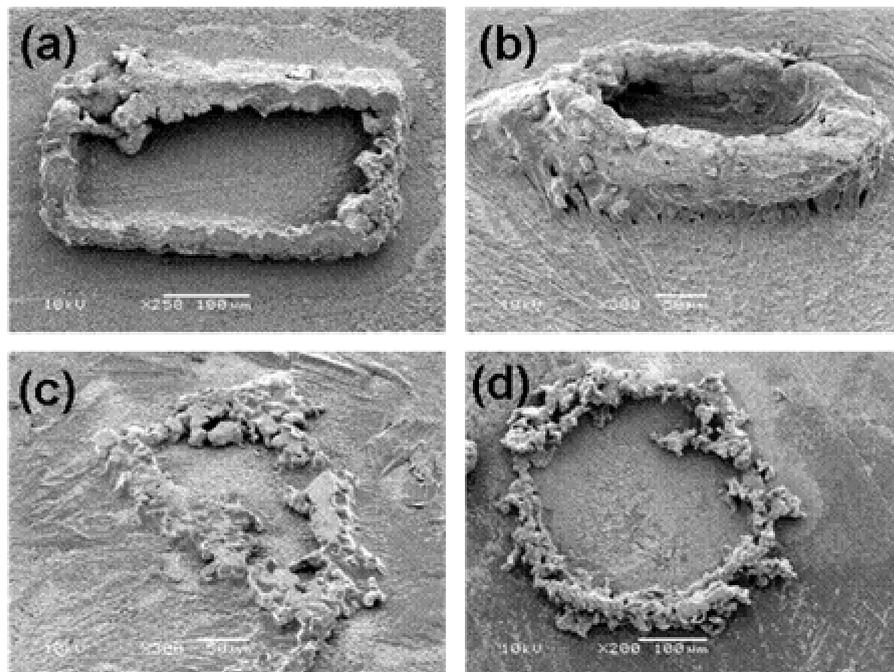


Fig. 7. SEM images of: (a) rectangular, (b) elliptical, (c) triangular, and (d) hexagonal micro-structures, deposited using stepping displacement of the tip.

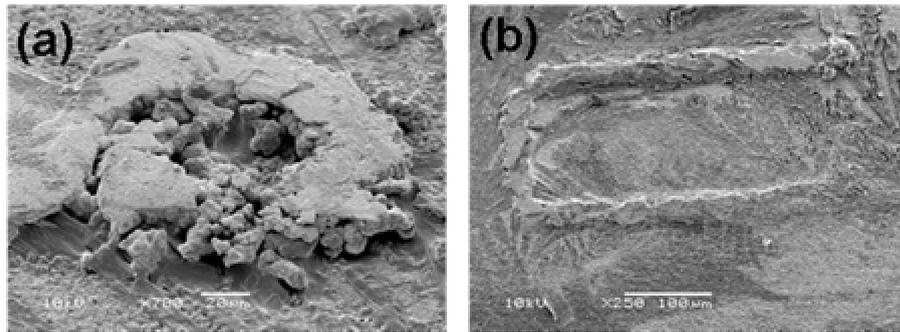


Fig. 8: SEM images of: (a) elliptical, and (b) and rectangular micro-structures, deposited using continuous displacement of the tip at a maximum displacement speed of 20 $\mu\text{m/s}$.

CONCLUSION

A synchronized linear displacement method was proposed to allow LECD fabrication of ASHAR micro-structures. The proposed synchronized electrode displacement was implemented using stepping and continuous displacement algorithms. The stepping displacement algorithm resulted in a successful deposition of ASHAR structures but with rough structure boundaries, especially those with dimensions close to the electrode tip. The continuous displacement algorithm produced structures with smoother boundaries, but since it was executed at the maximum tip travel speed of 20 $\mu\text{m/s}$ the deposition had a very slow growth rate. Faster growth rate is expected at a slower tip displacement speed.

ACKNOWLEDGEMENT

The investigator would like to express his sincere appreciation to the Research Affairs at the United Arab Emirates University for the financial support of this project under fund grant # 01-04-7-11/04. The investigator would also like to express his gratitude to the technical services of the Central Laboratories Unit for providing the SEM images of deposited samples.

REFERENCES

- Bard, A.J., O.E. Huesser, and D.H. Craston, "High Resolution Deposition and Etching in Polymer Films," U.S. Patent 4,968,390, 1990.
- Bard, A.J. and R. Faulkner, *Electrochemical Methods: Fundamentals and Applications*, Wiley, New York (1980).
- Dean, R.N., Jr., P.C. Nordine, and C.G. Christodoulou, "3-D Helical THz Antennas," *Microwave and Optical Technology Letter*, vol. 24, No. 2, pp. 106-111, Jan. 2000.
- Dizon, R., H. Han, A.G. Russell, and M.L. Reed, "An Ion Milling Pattern Transfer Technique for Fabrication of Three-Dimensional Micromechanical Structures," *Journal of Microelectromechanical Systems*, vol. 2, no. 4, pp. 151-158, December 1993.
- Durney, L.J., *Electroplating Engineering Handbook*, 4th edition, New York: Van nostrand-Reinhold, 1984.

- El-Giar, E.M. and D.J. Thomson, "Localized Electrochemical Plating of Interconnects for Microelectronics," Proc. IEEE Conf. on Communications, Power and Computing WESCANEX 97, pp. 327-332, 1997.
- El-Giar, E.M., R.A. Said, G.E. Bridges, and D.J. Thomson, "Localized Electrochemical Deposition of Copper Microstructures," Journal of the Electrochemical Society, vol. 147, pp. 586-591, 2000.
- Havemann, R.H. and J.A. Hutchby, "High-Performance Interconnects: An Integration Overview," Proceedings of the IEEE, vol. 89, no. 5, pp. 586-601, May 2001.
- Hunter, I.W., S.R. Lafontaine, and J.D. Madden and, "Three-Dimensional Microfabrication by Localized Electrochemical Deposition and Etching," U.S. Patent 5,641,391, 1997.
- Gelchinski, M.H., L.T. Romankiw, D.R. Vigliotti, and R.J. Von Gutfeld "Laser Enhanced Jet-plating and Jet-etching: High-Speed Maskless Patterning Method," U.S. Patent 4,497,692. February 1985.
- Gebhard, M. and W. Benecke, "Microtechnologies for Microscaled Robots and Components," Proc. 1995 INRIA/IEEE Sump. On Emerging Technologies and Factory Automation, pp. 9-20, 1995.
- Heb, C., K. Borgwarth, C. Ricken, D.G. Ebling, and J. Heinze, "Scanning Electrochemical Microscopy: Study of Silver Deposition on Non-conducting Substrates," Electrochimica Acta, vol. 42, nos 20-22, pp. 3065-3073, 1997.
- Ikuta, K., S. Maruo, and S. Kojima, "New Micro Stereo Lithography for Freely Movable 3D Micro Structures," Proceedings Of IEEE International Workshop on Microelectromechanical Systems (MEMS'98), pp. 290-295, 1998.
- Ikuta, K., K. Hirowatari, and T. Ogata, "Dimensional Micro Integrated Fluid Systems (MIFS) Fabricated by Stereo Lithography," Proc. IEEE Micro Mechanical Systems, pp. 1-6, 1994.
- Jansson, A., G. Thornell, and S. Johansson, "High Resolution 3D Microstructures Made by Localized Electrochemical Deposition of Nickel," Journal of the Electrochemical Society, vol. 147, pp. 1810-1817, 2000.
- Kikuchi, T., M. Sakairi, and H. Takahashi, "Three-dimensional Microstructure Fabrication with Aluminum Anodizing, Laser Irradiation, and Electrodeposition," Journal of the Electrochemical Society, vol. 150, no. 9, pp. C567-C572, 2003.
- Madden, J.D. and I.W. Hunter, "Three-Dimensional Microfabrication by Localized Electrochemical Deposition," Journal of Microelectromechanical Systems, vol. 5, pp. 24-32, 1996.
- Madiu, M.J., Fundamentals of Microfabrication, Boca Raton, FL: CRC Press, 1997.
- Müller, A.D., F. Müller, and M. Hietschold, "Localized Electrochemical Deposition of Metals using Micropipettes," Thin Solid Films, vol. 366, pp. 32-36, 2000.
- Odom, T.W., J.-L. Huang, P. Kim, and C.M. Lieber, "Atomic Structures and Electronic Properties of Single-walled Carbon Nanotubes," Nature, vol. 391, pp. 62-64, 1998.

- Paunovic, M. and M. Schlesinger, *Fundamental of Electrochemical Deposition*, John Wiley & Sons, Inc., New York (1998).
- Rai-Choudhury, P., *Handbook of Microlithography Micromachining and Microfabrication*, SPIE Optical Engineering Press 1-2, 1997.
- Rebeiz, G.M., L.P.B. Katehi, W.Y. Ali-Ahmad, G.V. Eleftheriades, and C.C. Ling, "Integrated Horn Antennas for millimeter-Wave Applications," *IEEE Antennas Propagation Magazine*, vol. 34, No. 1, pp. 7-16, Feb. 1992.
- Reynaerts, D., P-H. Heeren, and H.V. Brussel, "Microstructuring of Silicon by Electro-discharge machining (EDM) – part I: Theory," *Sensors and Actuators A*, vol. 60, pp. 212-218, 1997.
- Robertson, J.K. and K. D. Wise, "A Nested Electrostatically-Actuated Microvalve for an Integrated Microflow Controller," *Proc. IEEE Micro Mechanical Systems*, pp. 7-11, 1994.
- Said, R.A., "Localized Deposition of Copper Microstructures (II): Artifacts of Electrode-Tip Geometry," *The Electrochemical Society International Semiconductor Technology Conference 2001*, Shanghai, China, pp. 18-24, May 27-30, 2001.
- Said, R.A., "Localized Deposition of Copper Microstructures (I): Modeling of Shape Formation," *The Electrochemical Society International Semiconductor Technology Conference 2001*, Shanghai, China, pp. 9-17, May 27-30, 2001.
- Said, R.A., "Microfabrication by Localized Electrochemical Deposition: Experimental Investigation and Theoretical Modeling", *Nanotechnology*, vol. 14, pp. 523-531, 2003.
- Said, R.A., "Adaptive Tip-Withdrawal Control for Reliable Micro-fabrication by Localized Electro-deposition," *Journal of Microelectromechanical systems*, November, 2003.
- Said, R.A., "Localized Electro-Deposition (LED): The March Toward Process Development", *Nanotechnology*, vol. 15, pp. S649-S659, 2004.
- Said, R.A., "Localized Deposition of Copper Microstructures (II): Artifacts of Electrode-Tip Geometry," *The Electrochemical Society International Semiconductor Technology Conference 2001*, Shanghai, China, pp. 18-25, May 27-30, 2001.
- Said, R.A., "Localized Deposition of Copper Microstructures (III): Controlling Shape Formation," *The Electrochemical Society International Semiconductor Technology Conference 2001*, Shanghai, China, pp. 26-34, May 27-30, 2001.
- Smith, H.I. and H.G. Craighead, "Nanofabrication," *Physics Today*, pp. 24-30, February 1990.
- Sun, X.-O., T. Masuzawa, and M. Fujino, "Micro Ultrasonic Machining and its Applications in MEMS," *Sensors and Actuators A*, vol. 57, pp. 159-164, 1996.
- Timp, G., *Nanotechnology*, Springer-Verlag, New York, pp. 161-206, 1999.
- Van Der Schoot, B., M. Biollat, and N. De Rooij, "Micro-Instruments for Life Science Research," *I.E.E.E. Trans. Instrum. Meas.*, vol. 50, no. 6, pp. 1538-1542, 2001.
- Van Dyke, L.S., C.J. Brumlik, and C.R. Martin, "UV Laser Ablation of Electronically Conductive Polymers," *Synthetic Metals*, vol. 52, pp. 299-304, 1992.

Wallenberger, F.T., P.C. Nordine, and M. Boman, "Inorganic Fibers and Microstructures Directly from the Vapor Phase," *Composites Science and Technology*, vol. 51, pp. 193-212, 1994.

Wildoer, J.W.G., L.C. Venema, A.G. Rinzler, R.E. Smalley, and C. Dekker, "Electronic Structure of Atomically Resolved Carbon Nanotubes," *Nature*, vol. 391, pp. 59-62, 1998.

Yeo, S.H. and J.H. Choo, "Effects of Rotor Electrode in the Fabrication of High Aspect Ratio Microstructures by Localized Electrochemical Deposition", *Journal of Micromechanics and Microengineering*, vol. 11, pp. 435-442, 2001.

Yeo, S.H., J.H. Choo, and K.H.A. Sim, "On the Effects of Ultrasound Vibration on Localized Electrochemical Deposition", *J. Micromechanics and Microengineering*, vol. 12, pp. 271-279, 2002.

Yeo, S.H., J.H. Choo, and K.S. Yip, "Localized Electrochemical Deposition - the Growth Behavior of Nickel Micro-columns," *Proceedings of SPIE*, vol. 4174, pp. 30-39, 2000.