MICROFACTORY CONCEPT WITH BILEVEL MODULARITY

By
EMRAH DENİZ KUNT

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Philosophy of Doctorate

SABANCI UNIVERSITY
Fall 2011

MICROFACTORY CONCEPT WITH BILEVEL MODULARITY

APPROVED BY:

ASIF ŞABANOVİÇ ....
(Dissertation Advisor)

MAHMUT FARUK AKŞİT ....

AYHAN BOZKURT ....Ayhan Bozkurt

KEMALETTİN ERBATUR ....Kemaletti

GÖNEN EREN ....

DATE OF APPROVAL: 31.01.2012

ABSTRACT

There has been an increasing demand for miniaturization of products in the last decades. As a result of that, miniaturization and micro systems have become an important topic of research. As the technologies of micro manufacturing improve and are gradually started to be used, new devices have started to emerge in to the market. However, the miniaturization of the products is not paralleled to the sizes of the equipment used for their production. The conventional equipment for production of microparts is comparable in size and energy consumption to their counterparts in the macro world. The miniaturization of products and parts is slowly paving the way to the miniaturization of the production equipment and facilities, enabling efficient use of energy for production, improvement in material resource utilization and high speed and precision which in turn will lead to an increase in the amount of products produced more precisely. These led to the introduction of the microfactory concept which involves the miniaturization of the conventional production systems with all their features trying to facilitate the advantages that are given above.

The aim of this thesis is to develop a module structure for production and assembly which can be cascaded with other modules in order to form a layout for the production of a specific product. The layout can also be changed in order to configure the microfactory for the production of another product. This feature brings flexibility to the system in the sense of product design and customization of products. Each module having its own control system, is able to perform its duty with the equipment placed into it. In order to form different layouts using the modules to build up a complete production chain, each module is equipped with necessary interface modules for the interaction and communication with the other process modules. In this work, the concept of process oriented modules with bilevel modularity is introduced for the development of microfactory modules.

The first phase of the project is defined to be the realization of an assembly module and forms the content of this thesis. The assembly module contains parallel

kinematics robots as manipulators which performs the assigned operations. One of the most important part here is to configure the structure of the module (control system/interface and communication units, etc.) which will in the future enable the easy integration of different process modules in order to form a whole microfactory which will have the ability to perform all phases of production necessary for the manufacturing of a product.

The assembly module is a miniaturized version of the conventional factories (i.e. an assembly line) in such a way that the existing industrial standards are imitated within the modules of the microfactory. So that one who is familiar with the conventional systems can also be familiar with the construction of the realized miniature system and can easily setup the system according to the needs of the application. Thus, this is an important step towards the come in to use of the miniaturized production units in the industry. In order to achieve that kind of structure, necessary control hardware and software architecture are implemented which allows easy configuration of the system according to the processes. The modularity and reconfigurability in the software structure also have significant importance besides the modularity of the mechanical structure.

The miniaturization process for the assembly cell includes the miniaturization of the parallel manipulators, transportation system in between the assembly nodes or in between different modules and the control system hardware. Visual sensor utilization for the visual feedback is enabled for the assembly process at the necessary nodes. The assembly module is developed and experiments are realized in order to test the performance of the module.

ÖZET

Son yıllarda ürünlerin minyatürleştirilmesi yönünde gittikçe artan bir eğilim oluşmaktadır. Bunun sonucu olarak da minyatürleştirme ve mikro sistemler önemli bir araştırma alanı olarak ortaya çıkmaktadır. Mikro üretim tekniklerinin gelişmesi ve giderek kullanılmaya başlamasıyla yeni cihazlar markette yerini almaktadır. Fakat ürünlerdeki minyatürleşme üretimde kullanılan ekipmanların boyutlarıyla paralel ilerlememektedir. Mikro parçaların üretiminde kullanılan konvansiyonel ekipmanlar makro boyutlarda karşılık gelen ekipmanlarla boyut ve enerji tüketimi açısından karşılaştırabilir durumdadır. Ürünlerin ve parçaların minyatürleştirilmesi yavaş yavaş üretim ekipmanlarının ve tesislerin de minyatürleştirilmesine sebep olmaktadır. Bu şekilde üretim için gerekli enerji tüketiminin verimli kullanılması sağlanabilmekte, materyal ve kaynak kullanımı geliştirilebilmekte, üretim hızları ve hassasiyetleri de arttırılabilmektedir. Bütün bunlar bahsedilen avantajları olası kılan ve konvansiyonel üretim sistemlerinin minyatürleştirilmesini içeren mikro fabrika kavramının ortaya çıkmasına neden olmuştur.

Bu tezin amacı; üretim ve montaj için farklı konfigürasyonlar oluşturabilmek ve her bir konfigürasyonun değiştirilerek farklı bir ürünün üretimi için yeni bir yerleşim planı oluşturulabildiği bir modül yapısı geliştirmektir. Bu özellik; ürün tasarımı ve ürünlerin özelleştirilebilmesi açısından sisteme esneklik kazandırmaktadır. Her bir modül kendi kontrol sistemiyle ve içerisine yerleştirilmiş ekipmanlar ile atanan görevi yerine getirebilmektedir. Farklı yerleşim planları oluşturabilmek ve tam bir üretim zinciri kurabilmek için için her bir modül diğer süreç modülleriyle iletişim kurabilmek için gerekli arabirimleri içermektedir. Bu çalışmada, mikro fabrika kavramı geliştirilmesi yönünde iki seviye modülerlikli süreç odaklı modül kavramı sunulmaktadır.

Projenin ilk fazı bir montaj modülünün gerçeklenmesi olarak tanımlanmış ve bu tezin içeriğini oluşturmaktadır. Montaj modülü manipulator olarak atanan operasyonları gerçekleştirmek üzere minyatür paralel kinematik robotları içermektedir. Montaj

modülü varolan konvansiyonel fabrikaların minyatür bir versiyonudur (ör: montaj hattı) ve varolan endüstriyel standartlar mikro fabrika modülleri içerisinde benzeştirilmiştir. Bu şekilde konvansiyonel sistemlere aşina bir kişi minyatür sistemin kurulumuna da aşina olmakta ve uygulama gereksinimlerine göre sistemi kolayca kurabilmektedir. Böylece minyatür sistemlerin endüstride kullanılması yönünde önemli bir adım atılmış olunmaktadır. Böyle bir yapının oluşturulabilmesi için gerekli kontrol donanım ve yazılım mimarisi sistemin süreç gereksinimlerine göre kolayca konfigüre edilebileceği bir şekilde gerçeklenmiştir. Mekanik yapıdaki modülerliğin yanısıra yazılım yapısındaki modülerlik ve yeniden yapılabilirlik de bu açıdan büyük önem arz etmektedir.

Montaj hücresinin minyatürleştirme süreci sistemde kullanılacak paralel manipülatörlerin, farklı montaj istasyonları veya farklı süreç modülleri arasındaki taşıma sisteminin ve kontrol sistem donanımının da minyatürleştirme işlemlerini içermektedir. Gerekli görsel geribildirimi sağlamak amacıyla montaj süreci için gerekli noktalarda görü sistemi kullanımı gerçekleştirilmiştir. Montaj modülü geliştirilmis ve performans testleri için sistemde deneyler gerçekleştirilmiştir.

“To my family”

ACKNOWLEDGEMENTS

It’s been a real long journey. And it is a great pleasure to express my gratitude to all the people who has accompanied me through that journey. In particular, I would like to thank my advisor, Asif Sabanovic, for his patience, encouragement and confidence for all these years.

I wish to thank the members of my PhD committee; Mahmut Faruk Akşit, Ayhan Bozkurt, Kemalettin Erbatur and Gönen Eren for their interest in my work.

I would like to thank Ahmet Teoman Naskali, my closest colleague and friend who has great effort and help for the realization of this work. I also would like to thank him for helping me get through the difficult times with the fun environment and entertainment that he provided. I wish to thank Kazım Çakır, who has been a great friend and a partner from the very beginning with his everlasting support and constructive ideas. I’m also grateful to Osman Koç, for discussing and helping me for the extension of his ideas related to his thesis.

I would like to thank all my colleagues from Microsystems Laboratory and Mechatronics department, especially; Berk Çallı, Selim Yannier, Merve Acer, Utku Seven, Ahmet Can Erdoğan, İlker Sevgen, Zeynep Temel, Kaan Can Fidan, Edin Golubovic, Tarık Edip Kurt, Mehmet Güler and Süleyman Tutkun,.

And I wish to thank the most special person, who has always been with me, tolerating and supporting during such a hard and painful period. Thanks to you.

Lastly, and most importantly, I wish to thank my parents, Nevin Kunt and Türker Kunt and my sister, Derya Kunt. They’ve supported me in all my actions and in every aspect of my life. To them I dedicate this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF ABBREVIATIONS

POM	Process Oriented Module
CNC	Computer Numerical Control
MTS	Micro Turning System
MEL	Mechanical Engineering Laboratory
MMS	MicroRobot Based Microassembly Desktop Stations
MEMS	Microelectromechanical Systems
DOF	Degrees of Freedom
CCD	Charge-Coupled Device
POM	Process Oriented Modul
AMMS	Advanced Modular Micro Production System
AAA	Architecture for Agile Assembly
CSEM	Centre Suisse d'Electronique et de Microtechnique
TUT	Tampere University of Technology
EPFL	École Polytechnique Fédérale de Lausanne
MMC	Micro Machine Center
GUI	Graphical User Interface
CNC	Computer Numerical Control
MTS	Micro Turning System
FPGA	Field Programmable Gate Array
PC	Personal Computer
HDL	Hardware Description Language
IC	Integrated Circuit
PWM	Pulse Width Modulation

1 INTRODUCTION

The trend towards miniaturization of devices in the last few decades has led to the introduction of products and devices with sizes becoming smaller and smaller day by day. These products are composed of many parts which need to be integrated in order to form a functional product. Since the sizes of these parts are getting smaller, the imbalance between the size of products and the size of the manufacturing system becomes remarkably large. Nowadays, the machines for the production of micro-scale products are using almost the same techniques with the conventional macro scale production systems in the context of required space and energy. In that context, there is need for the miniaturization of the production equipment and facilities in order to decrease the space requirements, reduce the required energy for the production, improve material resource utilization, provide high speed and high precision and acquire flexibility since the layout of the production system can easily be changed when compared to the conventional production systems.

The term “microfactory” was initially proposed by the Mechanical Engineering Laboratory (MEL) in Japan in 1990 for their small sized manufacturing and assembly system [1]. The microfactory can be defined as a small production system which is suitable for manufacturing of small parts and assembly of these parts in order to form a product. Besides their advantages which are indicated above, microfactories will also address the market needs by increasing the possibility for the production of more customized products while ensuring the precise manufacturing and production speed.

With the introduction of microfactories, the production activities can be limited to the minimum possible level, in a way to produce only necessary things when they are really needed with a quantity of minimum requirements at the place where they are requested. That production attitude is also considered to respond flexibly to diverse needs of individual consumers.

1.1 Background and Motivation

The miniaturization of products and the recent developments in microsystem fabrication technologies has led to the necessity for an assembly process for the formation of complex hybrid microsystems. Integration of microcomponents made up of different materials and manufactured using different micro fabrication techniques is still a primary challenge since some of the fundamental problems originating from the small size of parts, high precision necessity and specific problems of the microworld in that field are still not fully investigated. The necessity of the assembly process in the microworld requires flexible, modular, accurate mechanisms, which can manipulate different types of objects in order to realize the assembly operation. Assembly operation plays a key role for the formation of a product since different materials within the content of a product lead to different functionalities.

Several groups have conducted research to develop microassembly systems. Flexible micro robot based microassembly desktop stations (MMS) in which microassembly processes are carried out by automatically controlled micro robots are proposed in [2], [3], [4]. Design and development of a 6 degree of freedom robotic manipulator used in the assembly of three-dimensional MEMS microstructures is presented in [5] as the further development of the 5 DOF manipulator presented in [6], [7]. A vision based feedback control system used in the automation of microassembly of MEMS devices using that 6 DOF robotic manipulator is presented in [8]. A microassembly system consisting of a 4 DOF base unit, a 2 DOF top unit equipped with an illumination dome and 3 microscopes with CCD cameras located on a ring structure above the whole system for the automated assembly of bio-micro robots is presented in [9]. In [10] multi-manipulator cooperation for the execution of microassembly tasks by using different kinds of micro endeffectors under stereo microscopic vision system is introduced.

Within the context of micro assembly and micro manipulation, in Microsystems Laboratory, a versatile and reconfigurable inspection and handling system for mini/micro products and components manipulation and assembly is developed [11], [12]. Differing from the previously conducted research, the workstation is designed and realized as a research tool for investigation of problems in microassembly and micromanipulation processes. The workstation is shown in Figure 1-1. The first prototype of the microassembly workstation is completed in 2006 and explained within the context of the MSc study [13]. Extra requirements according to the necessities that are discovered during the testing of the first prototype has led to the development of another workstation with necessary enhancements which constitutes a part of this PhD study and details about the workstation is given in the following chapters of this thesis.

During the development phase of the workstation great know-how about high precision system design and control was acquired. Experiments realized on the workstation demonstrated that the results are promising in the sense of precision, accuracy and reliability. This motivation yields us to focus on design and development of necessary hardware and software for the field of high precision system design and control.

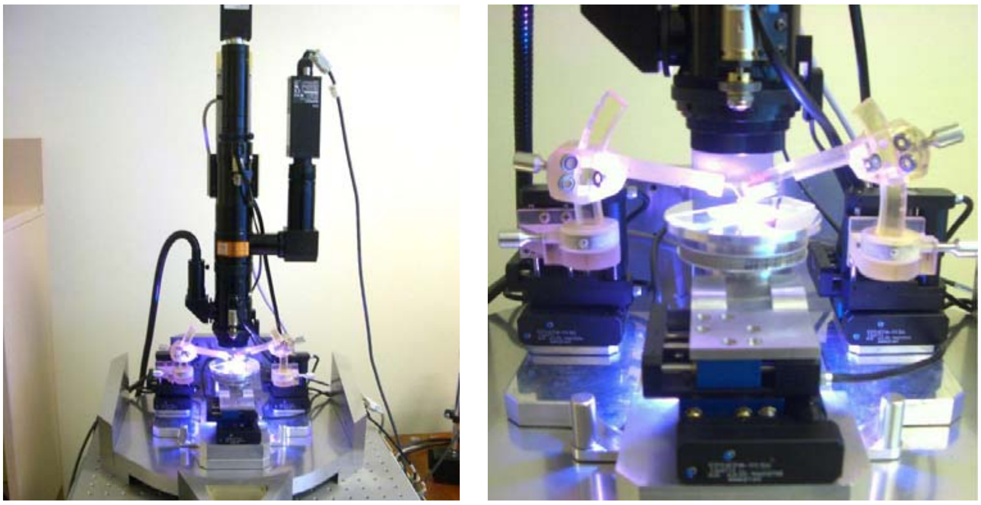


Figure 1-1 – Microassembly Workstation

The production equipments to produce the micro scale products are still using the same techniques, spending the same amount of space and energy to produce macro scale products. The micro factory concept arises from the notion that the better way to produce small parts is to use small size production systems. In other words, it is the concept of minimizing the production systems and processes to mate the products in size. Miniaturizing the manufacturing equipment will lead to improve resource utilization, production speed and precision, decrease space and energy consumption, noise, vibration and pollution enabling a more environmentally friendly factory concept. Additionally, it will lead to customization of products with on site manufacturing capabilities while ensuring the precise manufacturing of products. This motivates researchers towards development of small size production systems referred as micro factory.

Considering the relation between the part costs with the cost of the equipment, in order to reduce the cost of the part, low cost equipment with high speed is necessary which is possible with the concept of micro factory with the minimization of the distance and mass. By decreasing the equipment and factory size, layout spatial requirements will be less and production lines could be easily changed. Microfactories are also expected to take the production closer to the customer and revolutionize the manufacturing and assembly by addressing the market needs for more customized products while ensuring precise manufacturing of those products.

With this motivation through the development of microfactories, as a further step for the efforts in the field of high precision system design, a modular and reconfigurable microfactory concept is aimed to be developed.

1.2 Objectives

Within the context of this PhD study, in order to provide flexibility in product design and the customization of products, a bilevel modular robotic assembly cell which provides two layers of modularity is developed for advancing the microfactory concept. The module itself is used as a brick to establish a microfactory layout acting as a process module realizing one complete process within itself. The first layer of the modularity is achieved by using several modules, each implementing a different process (manufacturing, assembly etc.) and cascaded to each other in different configurations. Several layouts can be formed for the manufacturing of diverse products each having different kind and number of manufacturing and assembly modules in itself. Since the module is structured to be a process oriented module (POM), the flow of the process is reconfigurable also within the module which brings out the second layer of modularity. In that context, the components within the module can be located in different configuration which facilitates the reconfigurability of the process according to the products to be manufactured.

The robotic assembly module consists of all the mechanical units necessary for the assembly process, motion control hardware/software, vision system and main system supervision software. The assembly module also has parallel kinematic miniaturized robots (Delta robot, pantograph), serial kinematic manipulators, carrier units, sensors, stoppers, cameras and any necessary component for assembly. The performance of the system is tested with pick place experiments realized with miniaturized Delta robots (3 dof parallel kinematic robot) with the visual guidance supported by microscopic vision sensor located on the carrier unit. A graphical user interface is designed for the operator to easily realize the desired assembly tasks and control the system. The modularity of the process module is tested with the integration of a second Delta robot and the experiments are realized on the two manipulator system. The results that we obtained through the experiments are promising in the sense of the realization of such a modular microfactory concept and the initiative to use for real life applications in the industry.

1.3 Thesis Outline

The organization of this thesis is as follows; In Chapter 2, the state of the art in micromanufacturing and microfactory is given with the explanation of the developments in these fields and systems developed are given with details. In Chapter 3, the microfactory concept is explained with the advantages and disadvantages of the concept. The necessities and challenges for the development of the concept are described. The modular microfactory concept with the introduction of the bilevel modularity is described in details in Chapter 5. The details about the implementation of the assembly module with all its components and the performance analysis are shown. Details about the development of the assembly module are given in Chapter 6 and the thesis is concluded in Chapter 7 with discussions.

2 STATE OF THE ART IN MICROMANUFACTURING AND MICROFACTORY

The research in microfactories originates from early 1990's with the introduction of a desktop machining microfactory by the Micro Machine Center (MMC) and the Mechanical Engineering Laboratory (MEL) of the Agency of Industrial Science and Technology [1]. Apart from saving in space, another advantage of the miniaturization of the production equipment is low consumption of energy. According to the estimations of the originators of this first microfactory, a reduction of the size of a factor 10 can reduce the consumption of energy of a factor 100 [14]. The small size of the machines reduces the motion of the masses and thus saves energy while increasing the accuracy of movement. Since the system is more compact, the distances to be travelled are smaller which makes it possible to decrease the time of transport between the stations of production.

This first microfactory is a miniature chain of production units of machining and assembly for the manufacture of micro bearings with balls having diameters of 0.9 mm. All the parts of this bearing, except the balls, are manufactured in this microfactory. The machining units are: a micro lathe, a micro drill, a micro press, a transfer robot and a manipulation robot having the shape of two fingers. The dimensions of the whole system were 625 mm x 490 mm x 380 mm with a weight of 34 kg. The overall system is shown in Figure 2-1.

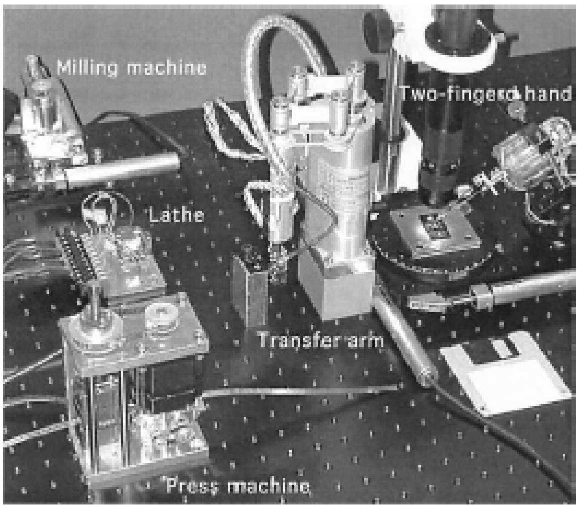


Figure 2-1 – The First Microfactory built by the laboratory MEL in Japan [14]

The idea of this project was centered on the miniaturization of the chain rather than the environmental conditions. It generates waste during machining and it is not compatible with a clean room. Modularity in this system was not a real issue as the main target was to create test platforms for the production of specific components or products in as small space as possible.

The project of microfactory of company OLYMPUS, developed since 1991, makes it possible to assemble a lens of a diameter of 1 mm, with a CCD camera and the support of the lens. Surface area of this installation is 500 mm x 350 mm [15].

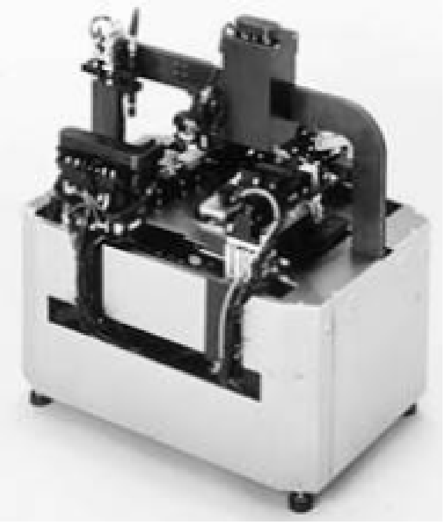


Figure 2-2 – Microfactory of Olympus [15]

The system described in [16] is a small microfactory which is able to machine micro pieces by electrochemistry. The microfactory for electrochemical process is shown in Figure 2-3. The dimensions of the system are approximately 0.9 m x 0.65 m. The workspace of the electrochemical process is of 5 x 5 mm^2. The resolution of machining is 20 µm with a depth of 300 µm. By changing the polarity of the electrodes, it is possible to make material deposition, just like removal by electro-erosion. The parts are carried on a carriage unit which can move in x and y directions with a speed of 50 mm/s.

For the assembly, this microfactory uses two robotic arms each one having 7 degrees of freedom and a conveyor. The robotic arms are 100 mm in length each having a 20 µm resolution by means of ultrasonic motors. Vacuum and an electromagnet are used according to the nature of the parts to be manipulated. The parts can move between these tools by means of a conveyor system made up of a grid of micro actuators each one has a size of 1 mm. This system can move components weighing up to 1 g with a speed of 30 mm/s. A CCD camera is used as the vision sensor in order to coordinate the operations.

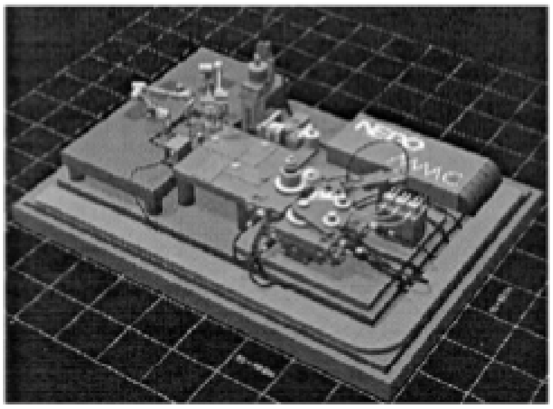


Figure 2-3 – Microfactory for Electrochemical Process

The Fraunhofer Institut for manufacturing engineering and automation IPA' in Stuttgart also developed a modular system of micro assembly [17], [18], [19], [20]. Dimensions of the modules are multiples of 100 x 100 mm^2. This system called AMMS for Advanced Modular Micro-Production System is composed of a standardized table on which it is possible to establish stations of assembly and shown in Figure 2-4. This insertion allows the transfer of energy, nitrogen, compressed air or vacuum between the base and the station of assembly. The parts are transferred between the stations by a conveyor system. The precision of positioning is 20 micrometers. A computerized interface makes it possible to design this assembly line virtually and to test it before its realization.

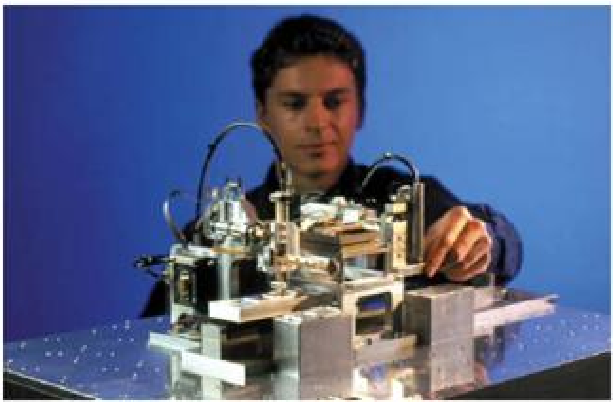


Figure 2-4 –AMMS by Fraunhofer Institute

Another project is the mini factory developed by Microdynamic Systems Laboratory at the Robotics Institute of Carnegie Mellon University. They developed a program named "Architecture for Agile Assembly" which helps to the design of the assembly line and makes it possible to simulate it, thus saving time of design. To test the performances of this system, a prototype of this mini factory was carried out which is made up of several modules. Each module comprises a working surface on which carriages move in XY planes with a positioning resolution of 200 nm by a system of electromagnet and air cushion. They collaborate with robotic manipulators which have 2 degrees of freedom. A vision system makes it possible to coordinate the whole system. This mini factory is able to assemble a microphone. As a whole, AAA was still not in micro scale, but presented most of the central ideas to be utilized in upcoming systems. [21], [22]

Since the late 1990s, the Institute of Production Engineering of Tampere University of Technology has also been involved with micro and desktop factory research. [23] The first prototype, called TOMI-mini factory, implemented sensor-guided assembly and disassembly of millimeter-scale planetary gear heads on a two-meter assembly line. This project is followed by ‘M4–Micro-Meso Mechanical Manufacturing’ project which aims to develop a modular box-type micro factory having module interfaces and connections to the outside world and upper-level systems and also clean room integration inside the microfactory modules. An assembly process of a cell phone loudspeaker with the size of 10.9 x 7.4 x 2 mm with a weight of 1 g is implemented with an assembly tolerance of ±0.2 mm. A Delta robot (Pocket Delta from CSEM) is used as the manipulator in the assembly module which is shown in Figure 2-6.

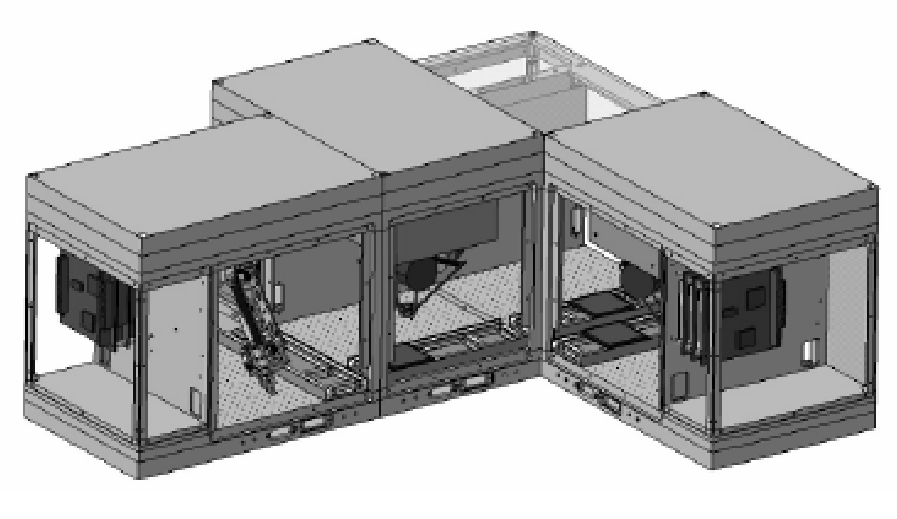


Figure 2-5 - M4 Modules assembled to form a microfactory

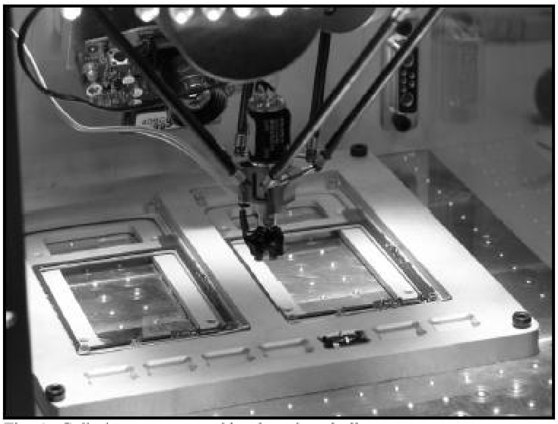


Figure 2-6– Assembling loudspeakers to the cell phone covers

As the extension of the M4 project, the TUT microfactory concept and its application for personalized manufacturing of a medical implant is introduced [24]. The system is configured in a construction kit type modular concept with the integrated clean room system and the integrated control system with a user interface. The TUT microfactory for a medical implant is shown in Figure 2-7. The implant microfactory includes a laser lathe with an on-line visual inspection system, a mini-sized manipulator for component handling and an ultrasonic washing system [25]. The modules developed are task oriented modules (TOMs) having only single task within the module.

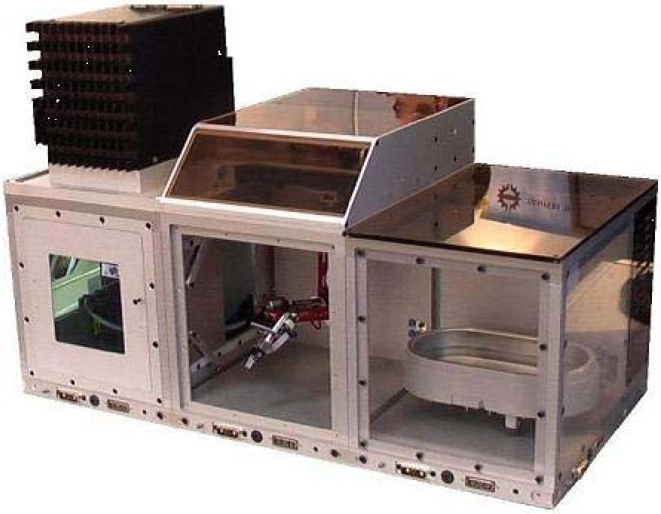


Figure 2-7 – TUT Microfactory for a medical implant

Another modular and miniaturized clean room system, so called Pocket-Factory, for assembly of small MEMS components was proposed by EPFL [26]. The size of the miniaturized clean rooms is 1 dm^3 for each module. The aim of the project is to combine the concepts of small assembly microfactory and the clean room environment.

Each module called microbox contains a small robot for the assembly and transfer tasks inside the microboxes between them. The modules have the same structure and are cascaded next to each other to form an assembly line. The microbox module is shown in Figure 2-8.

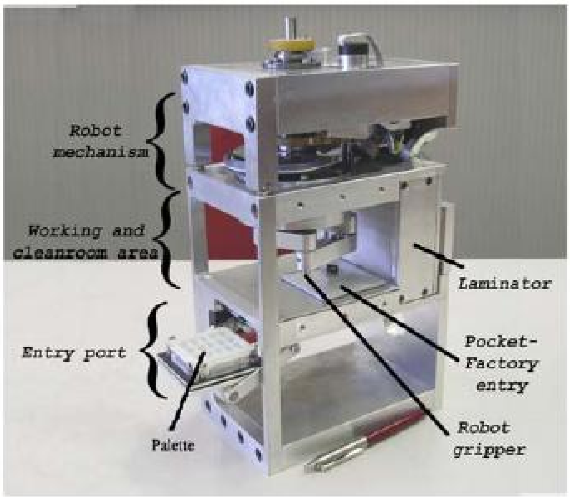


Figure 2-8 – The microbox module

3 MICROFACTORY

3.1 Introduction

The manufacturing systems required for the production of small parts must be more than a small version of a flexible manufacturing system. The fixturing limitations of small sized parts and the modularity considerations in a miniature production cell, requiring simple interconnections, recalibration and easy integration and disintegration of the production units are important concepts to be considered during miniaturization. In order to introduce a microproduct into the market there is need to develop materials, processes and production technologies for the massive production of the parts. The standardization issue has become an important concept in macro scale production which is also important for the concept of microfactories. In order to develop new products and new production techniques for the micro production concept, research has to be conducted on the material, high precision manufacturing, microassembly, handling units, microcomponents such as microsensors and microactuators to use for the miniaturized system actuation and in all the fields necessary.

The research on microfactory began with the idea of the necessity of miniaturizing the production equipment to match the microparts in size. The main goal of the research was saving energy and money. Towards such an effort, a microlathe smaller than a human palm as a part of the desktop microfactory was firstly developed by the Micro Machine Center (MMC) and the Mechanical Engineering Laboratory (MEL) of the Agency of Industrial Science and Technology in early 1990s. The developed microlathe was one cubic inch in size which was capable of cutting metal more accurately than a conventional lathe. It was the first unit of the desktop factory and the development of such a miniaturized machine became the driving force behind for further research in the field microfactories and microproduction units. After the development of the microlathe, the first prototype of the microfactory is developed as a full production line to produce small ball bearings by MEL.

3.2 The concept of microfactory

The concept behind the microfactories is the better way to produce a small part is to use a small machine. The small size of the production units enables the material resource utilization and the saving of energy and space. The reduction in the noise, vibration and pollution since the size of the machines are smaller will makes the system more environmental-friendly. Nowadays, the production cost of the micro systems is really expensive. In order to reduce the production cost the cost of the machines should be reduced and the speed of production should be increased. The minimization of the distance and the travelling masses enables high speed production. The following diagram summarizes the concept for microfactory.

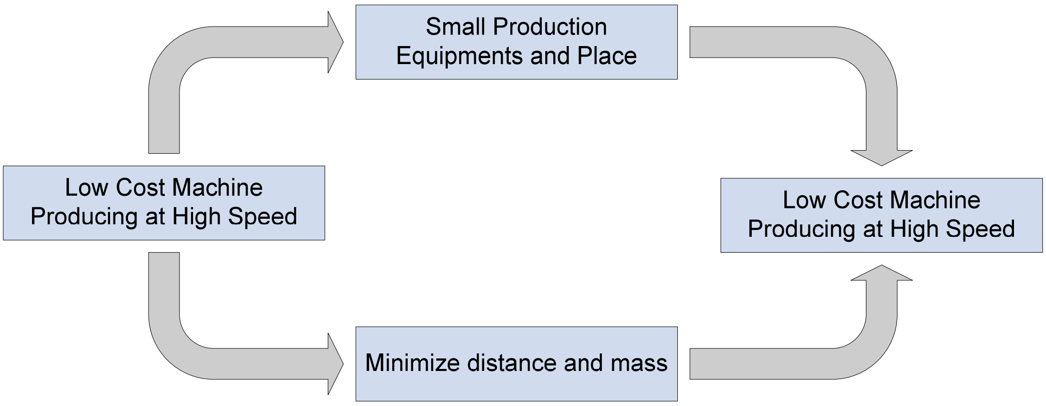

Small Production Equipments and Place
Low Cost Machine Producing at High Speed
Low Cost Machine Producing at High Speed
Minimize distance and mass

Figure 3-1 - Evolution of microfactory concept [27]

With the ongoing research towards the development of microfactories and miniaturized equipments, there will be a growing market for the microfactory. First of all, the existing machines and processes will be replaced by micro production units and newly developed micro processes. The estimated replacement market is around 1614M US$ by 2015 [27]. On the other hand, the research in this field will led to new ideas creating new fields of applications which will create a totally new market of products. The estimated new market is around 1875M US$ by the year 2015 [27].

3.3 Components of microfactory

The equipments to form a microfactory are small size equipments that a conventional factory contains. The requirements are the same as in the macro scale but differing in size, mass and accuracy. The modularity feature also appears in the design of the equipment and the software. The necessary components of a microfactory can be classified in two main layers; software and hardware. The main software and hardware components of a microfactory are shown in Figure 3-2.

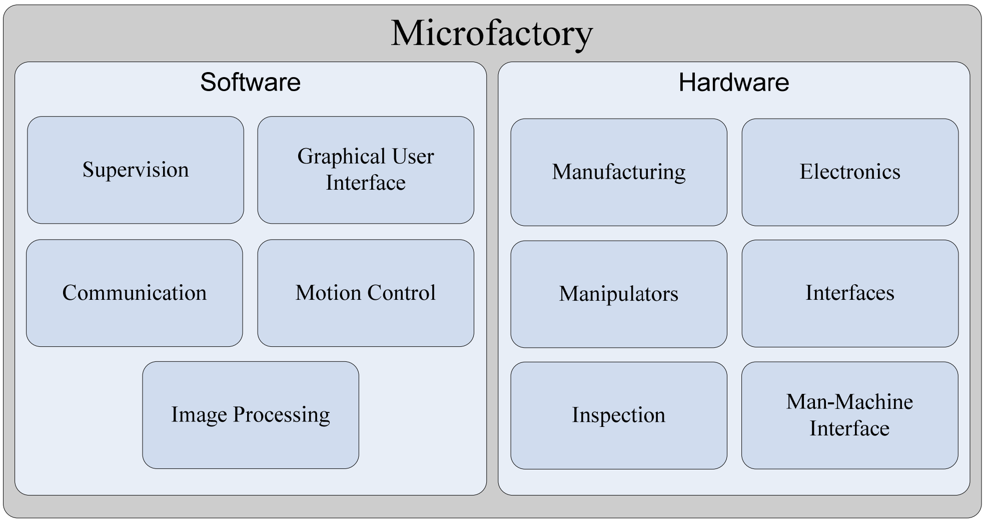


Figure 3-2 – Components of Microfactory

3.3.1 Software Components

- *Supervision* is the main structure of the software interconnecting and controlling every module according to the flow diagram of the system. This module should have a modular structure in order to allow modularity and reconfigurability of the microfactory modules.
- *Graphical User Interface* provides the interaction of the software and hardware units of the microfactory via human operators. The operator controls and observes all the states of the system using the GUI which is composed of system inputs and outputs as visual indicators. The GUI may run on a computer or a handheld device according to the needs.
- *Communication* software enables the interaction between all hardware and software units of the microfactory. The data between the units can be transferred between different units of the system using different communication protocols. Communication software is the unit that handles the data transfer according to the type of the protocol.
- *Motion Control* is the one of the most important components of the microfactory since the precision and accuracy of the actuators mostly depend on the control

performance. In order to achieve high precision and accuracy which is a must for the motion, the suitable algorithms are selected and implemented in the motion control software unit.

- *Image Processing* is necessary for the inspection of the processes, detection of the position and orientation of the parts, object recognition and for any other purpose where visual feedback is necessary. Since there are fixturing limitations as a result of the size of the parts, for the detection of the position and orientation of the parts a vision system is inevitable in a microfactory setup. Image processing software includes the algorithms and methods that are necessary to extract the necessary features and data for the system from the visual feedback supplied by the vision sensors.

3.3.2 Hardware Components

- *Manufacturing* components of a microfactory involves any type of miniaturized manufacturing system necessary for the production of the desired part. Micro lathe, micro drill, micro laser cutting, etc. can be given as examples of the manufacturing components.
- *Electronics* components can be examined as main control unit, interfaces and drive electronics. Main control unit is the processor board on which the whole software is running. Interfaces provide the connection between the peripheral electronics equipment and the main control unit, drive electronics is the interpreter between the control unit, actuators, peripheral equipments etc.
- *Manipulators* are robotic arms in several configurations with any number of degrees of freedom realizing the operations like transfer and assembly in the system. Serial or parallel kinematic structures can be selected according to the process necessities.
- *Inspection* units supply the necessary feedback data for the system. Vision systems for parts detection, product quality control, etc. and different kinds of sensors providing such data can be included into this category.
- *Interfaces* are the units providing the transaction of energy, air, vacuum and any necessary material for the flow of the production.
- *Man-Machine Interface* is the interaction device between the operator and the system. The operator can interfere and control the defined part of the system using the

man machine interface. Haptic devices and joysticks are mainly used as man-machine interface units.

3.4 Advantages of microfactory

The advantages of microfactories can be listed in various areas:

- *Space reduction:* The miniaturization of the manufacturing units and the components that are used in the production chain for microfactory requires less space occupation and facilitates smaller layouts. The modularity feature also enables efficient formation of the layouts and leads to the factory space reduction.
- *Cost reduction:* The reduction on the amount of space requirement allocated for the production facilities reduces the cost of space. Microfactories also enable material waste utilization which enables reduction in the waste material produced. Since the components used in microfactories are small machines, they consume less power which also leads to reduction of production cost.
- *Customization of products:* The modularity concept and the easy reconfiguration of the layout of the microfactories enable the customization of the products according to the needs of the customer. The flexibility in production achieved with the modularity feature makes it possible to produce different customized products using the same production facilities.
- *Modularization* is briefly the art of splitting up a product in a necessary and useful number of parts or sub parts, which give the producing company the ability to offer different variances of product by simply using different combination of sub products which are called product modules. The modularity concept brings flexibility to the production process since it makes it easier to reconfigure the system in a small amount of time when compared to conventional systems which makes it possible to produce customized products using the same units. Since the production units are smaller, reconfiguration of the product units is cheaper and faster which allows saving of money and time.
- *Flexibility* is one of the most important aspects of the microfactory. The modularity feature, small equipments, low space and energy requirements makes it easier to change the layout of the system when compared to the conventional systems in

terms of time, money and space. That allows the revolutionary feature of the microfactory which allows the realization of the production anywhere and anytime.

- *Inventory cost reduction:* Small production unit needs small amount of raw materials and consumable goods. So the stock size becomes smaller. It needs less space and less investment on raw materials.
- *Savings of energy:* In the conventional systems, manufacturing equipments are larger when compared to the size of the products they produce which results in a great amount of energy consumption. The miniaturization of the production equipment enables reduction in the energy consumption.
- *Environmental friendly.* A very important aspect of micro factory is its environmental friendliness. It saves energy, materials and resources which have positive impact on the environment. Miniaturization of the production equipment reduces the pollution, vibration and noise.
- *Response Time.* The modular design, compactness and small size of microfactory enables faster response when compared to conventional production units in the case of any change in the process or product. Currently, there is the problem of turbulent changes in the market where demands of the customer and technology are rapidly changing. A microfactory could be the best tool for manufacturing since easy reconfiguration of the whole system layout is possible.
- *Accuracy* of the production system is increased with the development of miniaturized production systems which have higher precision and accuracy resulting in production of precise parts and products.
- *Fast production* can be achieved since the mass of the parts are reduced and the distances to be traveled in the production are smaller when compared to the conventional systems which results in the reducing of the process chain in microfactory.

3.5 Miniaturization of Devices

The concept of miniaturization of production systems brings out the necessity of the whole systems and components used in the conventional macro systems to be miniaturized or replaced by new technologies. However, the miniaturization process includes many challenges since high precision is needed in every aspect of the process to achieve high accuracies. For the devices to achieve required precision and accuracy, mechanical and manufacturing tolerances are becoming significantly important. So the whole design and manufacturing process should be considered and designed carefully. Necessary components to build up a mechanism are not fully available for small sizes so custom made solutions should be realized in order to replace these components which appears as a challenge for the design of miniaturized devices.

Small dimensions make it possible for highly modular system design enabling the scalability and flexibility in the production layouts. It also makes the size adapted devices more robust against systematic errors caused by thermal expansion during operation and the dynamics of size scaled devices are better as a result of the scaling effects.

The size-adapted devices range from miniaturized precision robots to product specific assembly cells or production devices for the concept of a microfactory. In a microfactory according to the type of the product several types of processes take place like assembly, micro forming, micro turning and milling, etc. This brings out the necessity of miniaturization of the conventional machines or developing new technologies which can replace them. Miniaturized precision robots are considered mechanically as miniaturized versions of conventional robots based on well-known kinematic structures. These miniaturized manipulators are components of size adapted production systems which can be used for assembly processes in small sized production lines. Recent developments in the technologies such as emerging components like zero-backlash gears and highly dynamic micro-motors with integrated incremental encoders in the market, miniaturization of industrial robots is now possible. These scalable miniaturized structures lead to improved dynamic properties and process speed which are the result of their reduced dynamic masses.

In the following sections, examples of size adapted devices are given with details in two categories; manufacturing devices and assembly devices.

3.5.1 Manufacturing Devices

Size adapted manufacturing devices are mainly the miniaturized versions of the conventional machines that are used in production. These machines may have the same working principles with the macro scale versions. Additional to the small versions of the conventional machines, producing microproducts may necessitate new production techniques. Within the context of the microfactory research, several machines are developed and tested, examples of which are given in the following sections.

Micro lathe is the first machine developed within the concept of microfactory research. It is a part of the first microfactory developed by the Mechanical Engineering Laboratory (MEL) of the Agency of Industrial Science and Technology. The size of the lathe is slightly bigger than a cubic inch and weighs only 100 g. It has an XY linear stage driven by piezoelectric actuators. The main spindle motor has only 1.5 W rated power and the rotating speed is about 10,000 rpm. The micro lathe can cut brass with an accuracy of 1.5 μm roughness in the feed direction and 2.5μm roundness. The minimum diameter of the work achieved in the experiments was 60 μm. The lathe is equipped with numerical control system provided with high resolution motion control capacity.

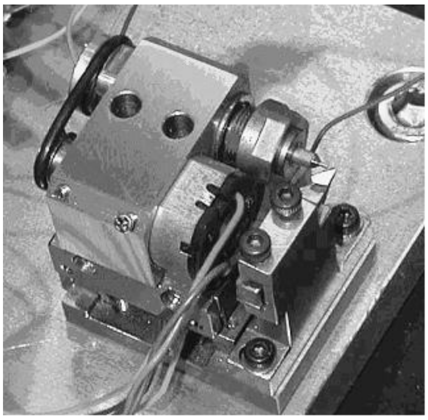


Figure 3-3 – Micro-lathe [14]

A micro milling machine is also developed within the content of desktop machining microfactory with a rated rotating speed up to about 20.000 rpm and performing surface cutting and drilling using end mill tools with a 3mm shank diameter.

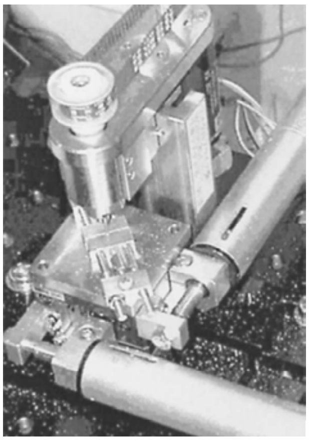


Figure 3-4 – Micro milling machine [14]

Another machine for drilling and milling operations has a 320,000-rpm air turbine spindle. The layout of the machine is 25 mm x 25 mm x 25 mm. The theoretically achievable speed is greater than 50 mm/s, the resolution is 0.1 µm, and the holding force is 10 N. High-speed miniature spindles are utilized that are required to obtain appropriate cutting velocities for the efficient cutting of metals. In order to collect force data for the experiments the machine is equipped with load cells. Three-dimensional features are machined and cutting force data, surface finish data, and machined feature profiles are presented in [28].

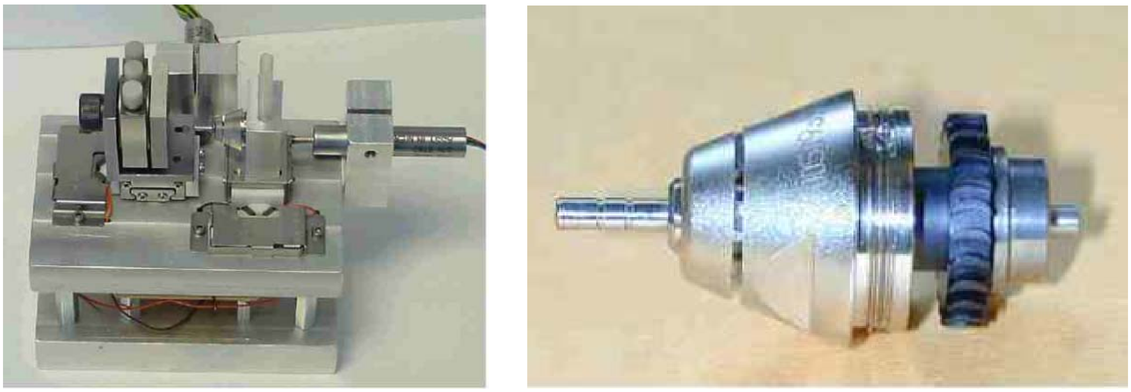


Figure 3-5 - (a) Two axis piezo-actuated testbed configuration and (b) 320,000 rpm air-turbine spindle [28]

The micro press machine developed by MEL has the dimensions of 111x66x170 mm and 100W rated power. It can generate a press load of about 3kN. The press speed and dead point of the press stroke can be numerically controlled. The press was used to produce the outer shell of the ball bearings that are produced in the first microfactory which are shown in Figure 3-7.

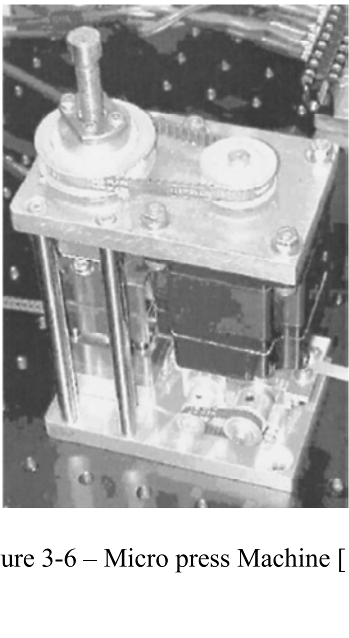


Figure 3-6 – Micro press Machine [14]

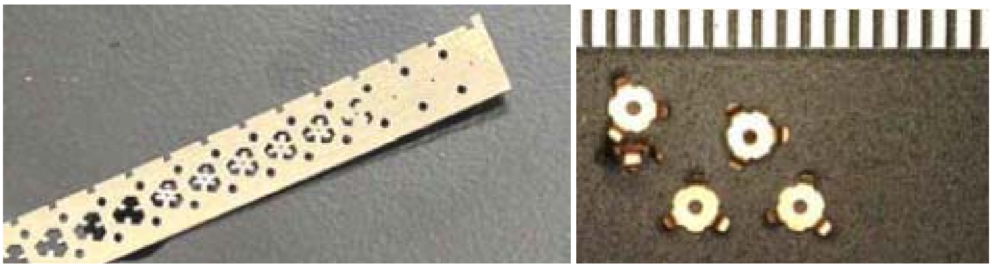


Figure 3-7 – Micro Press Machine Outputs

Tokatsu-Chiba Local Consortium developed a mini production system for small mechanical parts, which combines turning and grinding cells. Each cell occupies just a 200 mm square. The grinding cell was then integrated into a production line which is 1 m long, together with a cleaning unit, an inspection unit and transfer units. It is claimed that the floor space was reduced to 1/30 and energy consumption was reduced to 1/5 of conventional cylindrical grinding machines. Problems from lowered stiffness and machining power due to the miniaturization were overcome by new mechanical design and process control. [29]

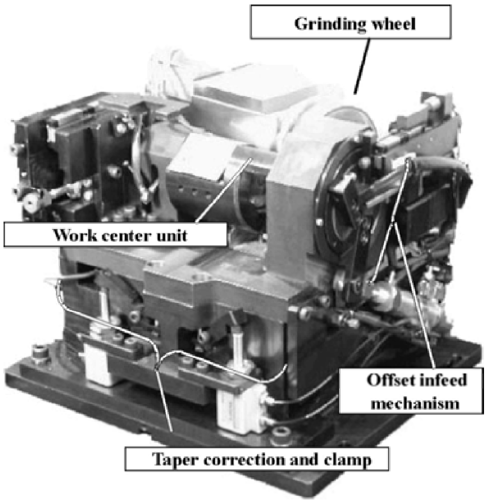


Figure 3-8 – Mini Grinding Cell

Many miniaturized production machines have already been utilized or commercialized in Japan, influenced by the early R&D projects on miniaturization of machines and production systems. One of the companies, The NANO Corporation has developed Micro Turning Systems. It is a palm-top precision lathe that has a machine base size of 150 x 100 mm with CNC with linear/circular interpolation. MTS is capable of machining brass with a surface roughness of 0.20 μm and a circularity of 0.19 μm.

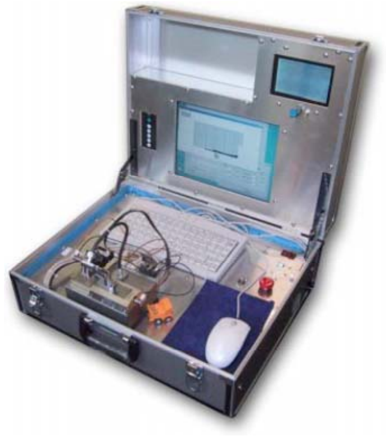


Figure 3-9 – Micro Turning System [30]

3.5.2 Manipulation Devices (Assembly Devices)

In a microfactory setting, additional to the production machines some transfer and assembly mechanisms are also needed since it is the case in conventional macro systems. There is a need to transfer the parts in and out of the production units, onto the carriage units and also as a nature of the most of the products; an assembly process. These processes are mainly performed by different configurations of serial and parallel manipulators according to the needs of the process.

Serial manipulators mainly consist of a number of rigid links connected with joints. The configuration of the serial manipulators is with revolute or prismatic joints and orthogonal, parallel and/or intersecting joint axes considering the simple kinematic solutions and manufacturing purposes. The main advantage of the serial manipulators is the large workspace. They can span a very large area when compared to the volume and occupied space by the manipulator. On the other hand, as a result of the serial structure of the mechanism, the precision is low since the error is accumulated through the end effector of the manipulator. They have low stiffness and as a result of the serial structure each actuator has to carry all the consecutive actuators in the kinematic chain. The serial manipulators are widely used in the conventional systems. Miniaturized versions are also developed for a variety of applications. The miniaturization of the actuators and the mechanical components necessary to build up a miniature robot has paved the way and additional to the systems developed towards the research in microfactory there are many commercially available manipulators and high precision miniature stages in the market with different actuation methods. The small sizes allow them to be configured as serial manipulators with very small footprints.

The micro transfer arm developed within the content of the first microfactory can be given as an example to the miniaturized serial manipulators. The micro transfer arm has a position accuracy of 20 μm within a workspace of 200 mm diameter circle and the two fingered micro hand can be driven in three translational motions in a range of 100x100x30 μm with a resolution of 1 μm. The maximum size of an object to be handled is 200 μm.

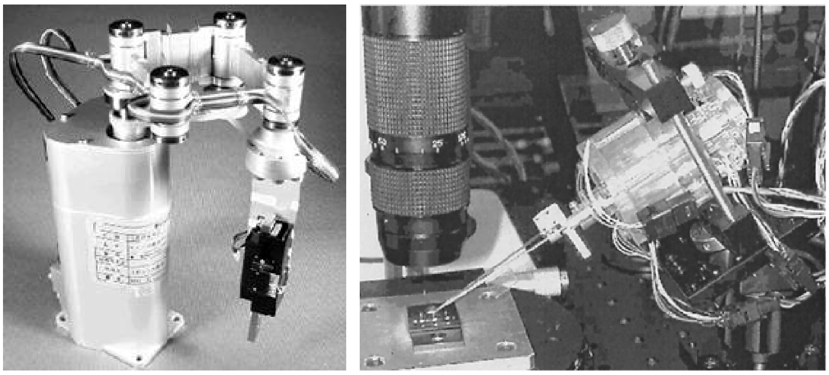


Figure 3-10 – a) Micro transfer arm b) Micro manipulator [14]

In the past several decades there has been a great interest towards parallel robots since these parallel structures possess several advantages over serial ones such as high stiffness, high accuracy, high payload-to-weight ratio, etc. Since the trend towards miniaturization has been increasing in recent years, there is a necessity for manipulators with high precision and accuracies. As a result of that, many parallel mechanisms with different number of degrees of freedom have been proposed. A fully parallel robot is a closed loop mechanism with an *n* dof end-effector connected to the base by *n* independent chains which have at most two links and are actuated by a unique prismatic or rotary actuator [33].

The main disadvantage of the parallel robots is the limited workspace. However, when precise handling of small particles is concerned, this may not be considered as a major problem since a small workspace is adequate for such applications.

Considering the advantages of parallel robots for high precision applications like microassembly, microinjection, etc. great effort has been put on the miniaturization of these robots in order to easily integrate into systems designed specifically for such applications. The Stewart Platform [31], originally proposed as a flight simulator platform, has been studied extensively and is widely used today with different variations in size available also in the market.

The limitation of the workspace is somehow solved by the introduction of the famous three-degree-of-freedom fully parallel Delta robot by Clavel, [32], which is dedicated to high-speed applications.

The Pocket Delta robot is a micro robot based on the parallel structure of the Delta Robot. It has been designed to perform microassembly tasks where high speed and high precision are needed in a reduced working space. The robot has 4 degrees of freedom, a repeatability of 5µm, payload of 20 grams and a workspace diameter of 80mm with z motion of 30mm. The integration of the miniaturized version of the Delta Robot, developed by CSEM, within a microfactory cell for assembly tasks is defined in [34].

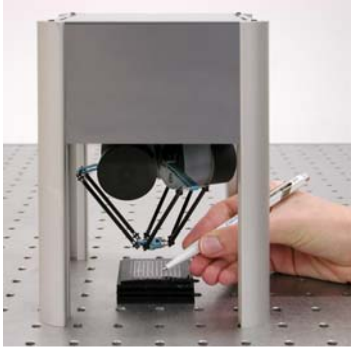


Figure 3-11 – Pocket Delta Robot

There are now commercially available parallel kinematics hexapods with six degrees of freedom. As an example PI (Physik Instrumente) offers several models of hexapods. The smallest one The M-810 Miniature Hexapod features high speed with direct drive torque motors up to 2.5 mm/sec for loads up to 5 kg. It has a travel range of 40 x 40 x 13 mm and rotation to 60 degrees with repeatability up to ± 0.5 µm. The dimensions of the hexapod are Ø 100 x 118 mm [35].

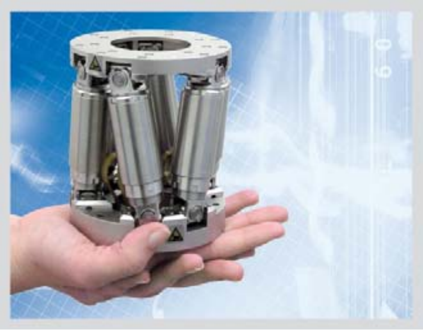


Figure 3-12 – Hexapod

MICOS Gmbh also developed several hexapods with different sizes and specifications and the smallest one is HP-140 with travel ranges of 32 x 32 x 12 mm in linear axes (xyz) and Rx, Ry 12°, Rz 20 ° in rotational axes. It has the maximum speed of 1mm/sec with dimensions 120mm height with a Ø 140 [36].

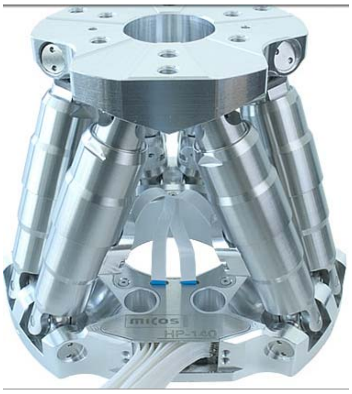


Figure 3-13 – MICOS HP-140

For the assembly needs of the Pocket-Factory developed in EPFL, each microbox has a small 4 degrees of freedom robot similar to a SCARA robot (x, y, z, and θ_z). It executes assembly and conveying tasks or collaborates with a high precision robot for more precise assembly. It is used to transfer parts inside each microbox and from one microbox to the next one. Moreover, it is used to open the door of the entry port. It operates in the clean room environment class ISO5. It has a workspace as a cylinder of 130 mm diameter and 20 mm height. The robot itself has a size of 100x100x200 mm. The 4 DOF small-scale parallel hybrid micro Scara robot shown in Figure 3-14 is explained in [26] and [37].

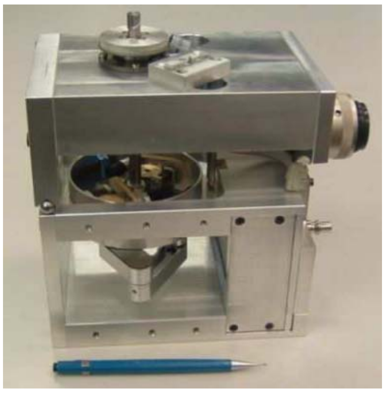


Figure 3-14 - The 4 DOF robot associated with a microbox

4 MODULAR MICROFACTORY CONCEPT

4.1 Introduction

The research of microfactory began in early 1990’s and mainly originated in Japan. First factories are mainly mini production systems where processes are not fully automated in which the tele-operation is achieved by low degrees of freedom manipulators. One of the most important features of the microfactory concept; modularity was not a real target at that time. The main aim of the researchers was to develop production platforms and devices as small as possible in order to save space and energy. However, the researchers began to focus on the modularity issue since the small size of the units that compose the microfactory makes the system more suitable to setup a modular structure which brings the flexibility in production facilities.

4.2 Design Requirements

The design requirements for a modular microfactory are more than the necessities to consider only for the miniaturization of the production system. In each unit of the system, modularity issue should be considered and the system specifications should be determined accordingly. The modularity issue mostly affects the design of the system components which are given in Section 3.3. In the following sections design requirements for a modular microfactory concept are explained in details.

4.2.1 Modularity

Modularity is one of the most important features of the microfactory concept and the units of the microfactory should be realized in a modular way in order to obtain the advantages that modularity provides. When the modularity is achieved, flexibility appears in the production process which enables producing different products by simply reconfiguring the production units or the layouts of the system which is cheaper and faster when compared to the conventional production systems. In that sense in the design of the microfactory concept modularity should be an important design criteria.

The modularity concept can be achieved by dividing the whole system into subunits which can be called the modules. The decision of splitting up the whole system in order to configure the modules is an important step for the microfactory concept to be generated. The modules should be developed in such a way that easy configuration of a complete production system can easily be generated by cascading the modules and forming an efficient layout for the production system.

The modules can mainly be realized in two different ways;

4.2.1.1 Task Oriented Modules (TOMs)

These modules are focused on the task and each of them contains only a single task unit such as manipulation (pick-place, assembly, etc.), inspection, machining (drill, lathe, milling, press, etc.), processing (heating, cooling, UV exposing, etc.), transferring, testing, etc. In order to realize a task inside the module, each module requires its own electronics and software. The modules should also have interfaces for the communication and interaction with the other task modules. Each module should contain all of these in order to work as a stand-alone unit so when cascaded with any other task oriented module it can perform its assigned task.

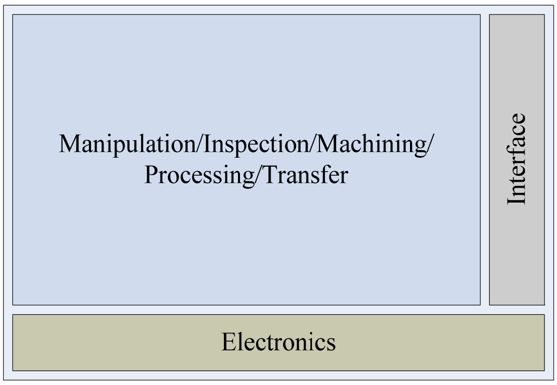


Figure 4-1 - Task Oriented Modules

4.2.1.2 Process Oriented Modules (POMs)

Rather than including a single task within itself, POMs can contain several tasks in order to perform a whole specific process. According to the defined process a POM can be configured to realize the process in different configuration of the tasks within itself. As a module, these modules also contain its own electronics and the interface as it is the case in task oriented ones.

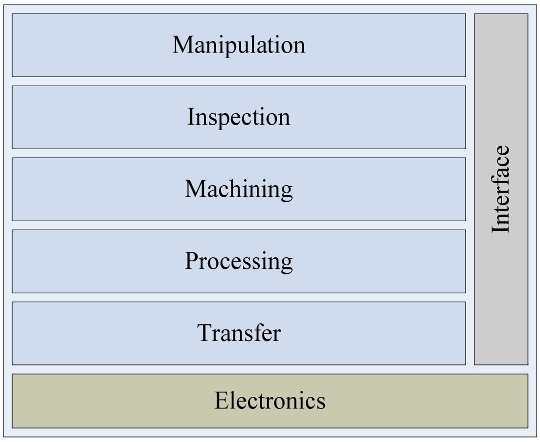


Figure 4-2 - Process Oriented Modules

The combination of the task units within the module allows it to realize a whole specific process like assembly. As an example, an assembly module may contain an inspection unit, a manipulation unit and necessary transfer units to carry the parts within these units.

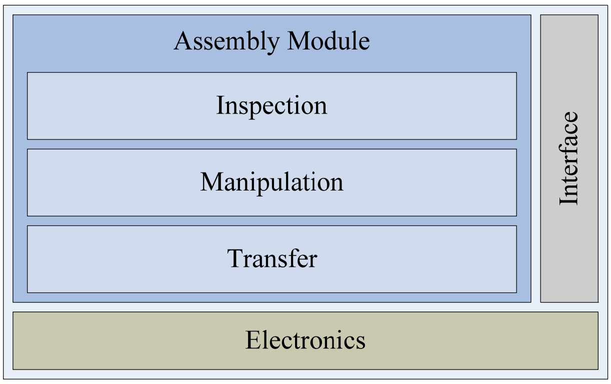


Figure 4-3 - Assembly Module (POM)

The microfactory module concept realized within the framework of this PhD work is a process oriented module (POM) rather than single task oriented ones. An assembly module containing the whole assembly operation for the defined task within the module is the first module to be realized structured as a POM. In order to conserve the modularity in every aspect of the microfactory concept, the modules should also be modular within itself (modularity within the module) so that for different kinds of assembly operations it can be configured with the replacement of manipulators, sensors and transfer mechanisms. That will ease the configuration of the microfactory for a specific product since for each task oriented module; a transfer mechanism is needed between the modules for the transfer of the product/sub-products. Each module has a minimum footprint, energy consumption and cost as a result of the interfaces, control unit and the module structure so that if the number of the modules used in the microfactory increases, the space and energy used also increase. In that context, the concept of the process oriented modules has the advantage of space and energy saving over the task oriented modules. POMs have also the advantage of time saving since the time spent in between the modules is reduced.

4.2.2 Task Units

Task units of a modular microfactory are the size adapted devices which are miniaturized in order to realize the operations necessary for the production process. These can be manufacturing devices or assembly devices examples of which are given in Section 3.5.

Miniaturization of devices requires development of a whole range of new miniature servo systems and measurement systems with very high accuracy and repeatability. High precision and small size necessity limits the selection of the actuators but with the recent developments in this area, there are now many small size actuators, precise measurement sensors and mechanical components suitable for high precision system design available in the market which makes the high precision miniaturized system design possible. Zero backlash gear heads, anti-backlash gears, small sized dc motors, brushless motors with integrated high resolution encoders, piezo actuators, strain gages or capacitive sensors for the measurement and high precision linear guides are all available commercially.

The following parameters mainly define the characteristics of a device which should be considered as design parameters to be achieved when designing a miniaturized machine;

- Workspace/Travel Range
- Precision/Accuracy/Repeatability
- Maximum Velocity
- Maximum Load

- Mass
- Operating Temperature/Voltage

These parameters should be determined before the design process since the components to be used in order to build the system should be selected accordingly. There may appear other parameters to be considered with respect to the process that the machine will realize. The production unit might be a laser micromachining unit for which the characteristics of the laser should also be considered as a design parameter for the development of the machine.

The manufacturing of the parts is another issue to be considered for the miniaturized devices. There will be the manufacturing and assembly intolerances which should be considered after the design process since they affect the performance of the devices. In that context high precision manufacturing of the parts is another issue to be considered for the realization of miniaturized devices.

4.2.3 Control System

A microfactory system is composed of many different functional components in order to facilitate the production process. For the functionality of the whole system working efficiently in order to perform its assigned duties, all the components should work in interaction with each other or independently to perform their function for the process flow. The main control system is the one that manages every functional unit, the interaction between them and their role in the process flow. For that reason, the control system structure should be carefully considered for the functionality and efficiency of the system as a whole.

The components of a microfactory are listed and explained in Section 3.3. Looking up from the system supervision and control side, a microfactory system may contain the following components;

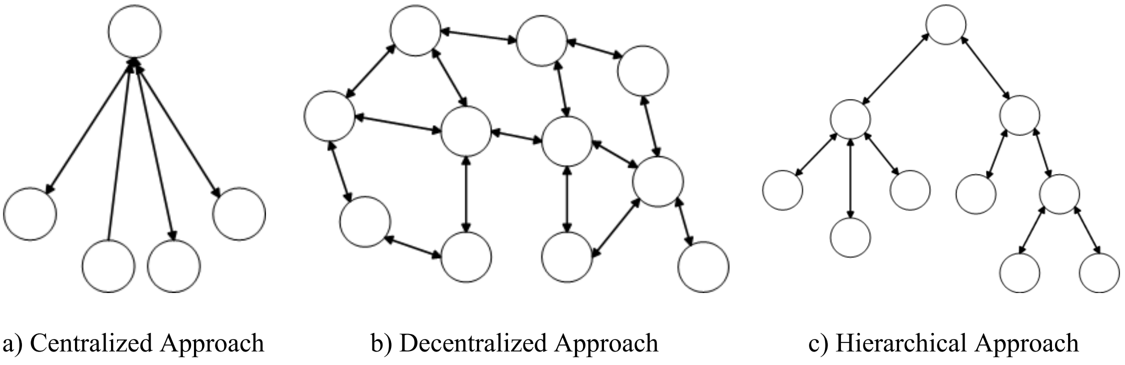
- Different type of actuators with associated drivers/controllers which are used in the manipulators, manufacturing devices, carriage units, etc.
- Sensors and measurement devices for the size adapted devices or to be used directly in the system
- Inspection system composed of different type of cameras, magnification systems like optical microscopes etc. if necessary.

- A control computer running in real-time for processing the data according to the related reference and sensor inputs coming from different modules of the system and generating desired outputs for the system
- A man machine interface for the human interference to the specified units of the system that exchanges information between the control computer and the operator
- A real-time communication network which can guarantee the maximum delay times and data losses in which the nodes can exchange information instantly or with negligible latency.
- A non-real-time communication network cannot guarantee the delays and losses so that delayed or lost messages in such a network cause degradation in quality rather than a total failure as it is in a real time network.
- A main supervision computer that has the main process control software running on it.

The control structure should be configured in such a way that all these components of the system can operate independent of but in interaction with each other and the main supervision or control computer. In order to assure the coordination of the components, the implementation of the control system should be realized according to the effective functionality of the system requirements.

The simplest approach is the centralized approach in which all the system components are connected to the powerful central computer and all the information is concentrated in that computer. This approach lacks advantage for the modularity and scalability issues. The decentralized approach mainly built upon the structure in which each module executes its own functions and exchanges the minimum information among each other. The data transfer is reduced but the stability of the system is hard to be guaranteed since it is hard to manage the system in which the functionality strongly depends on the coordinative behavior of each module in the system. In the distributed approach, the computational work is distributed between the modules and the main computer. Fast control loops and some simple operations are done on the modules and the issues related to the supervision of the system are realized by the main computer.

The complexity, the number of modules in the system, the necessary interconnectivity issues, data flow and processing and the main flow of operation in the desired microfactory are the important issues to be considered when building up the control structure of the system. These should be examined and the appropriate structure should be applied for the effective functionality of the system.


a) Centralized Approach

b) Decentralized Approach

c) Hierarchical Approach

Figure 4-4 – Control Structure Approaches [13]

4.2.4 Inspection System (Vision System)

The fixturing limitations as a result of the small size of the parts that are processed in the microfactory setup bring out the inevitable necessity of visual feedback for the position and orientation determination of the parts to be processed. Visual feedback is also used for quality control issues or even in the size adapted devices as a part of the production process by giving necessary feedback about the status of the operation. The feedback provided by the vision system must be precise and accurate enough to match the desired precision of the whole system. For that reason, additional to the hardware requirements, the image processing algorithms implemented to recognize the geometries, determining the position and orientation of parts, path generation for the manipulators, etc. must be selected and implemented effectively since they also affect the performance of the vision system.

Considering the high precision feedback necessity of the system as a whole, the parameters for the selection of a vision/inspection system can be listed as follows;

- *Magnification* is the measure of the size of the object in the image over the actual size of the object. The total magnification value is related to the optical system between the object and the camera, the camera coupling magnification and the scaling between the sensor size and the image size. The magnification value must be selected according to the smallest feature that is desired to be measured using the vision system.
- *Resolution* is the measurement of the ability of an optical system to distinguish the physical details of the object and reproduce these details in the image. It determines the

lower limit of the feature size to be detected and it represents the precision of the vision system for feature determination.

- *Working Distance* is the distance between the target object and the objective of the vision system. It is a critical parameter when online visual feedback is necessary for the operation that takes place under the vision system. It should be large enough to provide working space for the manipulators or any other operational system.
- *Depth of Field* is the amount of distance that allows the maintenance of acceptable image without refocusing. Narrow depth of field allows precise focusing on the planes which can also give depth feedback. On the other hand, when 3D geometry information is necessary narrow depth of field becomes a problem since it obstructs feature extraction in the unfocused parts of the object.
- *Field of View* is the measure of the visible area by the vision system. Field of view and magnification values are inversely proportional. According to the process necessities a vision system with fixed magnification or variable magnification can be the matter of choice. When fixed magnification is enough for the application, the field of view can be a critical feature in order to cover the whole workspace of the system.
- *Illumination* is the most critical factor for a vision system since the image is formed with light. The illumination technique must be carefully considered according to the desired functionality of the vision system since different illumination techniques may highlight different features of the target object.

5 DESIGN AND IMPLEMENTATION

In this section, design and implementation issues of the microfactory modules proposed in the context of this thesis are explained with respect to the design requirements listed in the previous section. The bilevel modularity concept introduced for the microfactories is explained. System supervision including the software structure and the electronics design considering the modular design issues is given in details. The assembly module developed for the verification of the bilevel modularity concept and the components of the assembly module; parallel manipulators, carriage units, vision system and the system as a whole are explained in the following sections.

5.1 Bilevel Modularity

Modularity can be defined as the ability to offer a variety of products by using different combinations of product modules which are the sub systems of the whole manufacturing system. Modularity should be implemented in such a way that the production process or production system can be reconfigured to produce a new product in a small period of time, without buying additional production equipment. This will fasten the response to the changes in demand of the customer and the market. The modularity concept is hard to implement in a conventional manufacturing system however, it is more suitable for the microfactory concept since it will be much cheaper and less time consuming to modify and change the manufacturing plant when compared to conventional plants.

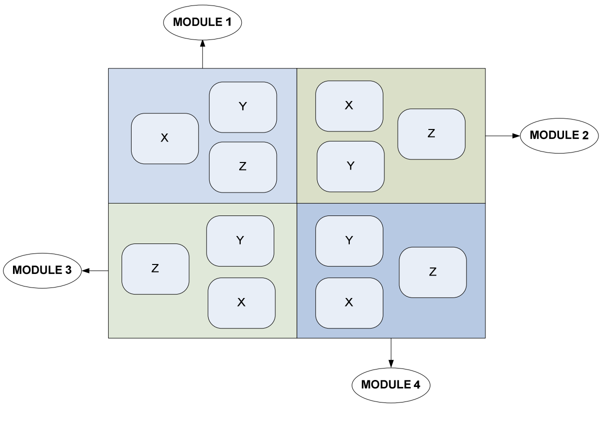


Figure 5-1 – System Modularity

In the microfactory concept the layout can be configured in such a way that equipping each module with sub modules configured to produce a sub product each of which will be used to form the final product. The configuration of each module may vary according to the needs. The concept of system modularity is illustrated in Figure 5-1. Considering the sub modules X, Y, Z as different machining and assembly units, each module is configured with different combination of these sub modules producing a product module. For the final product, four modules are cascaded and according to the complexity of the product the number of modules can be increased and the layout can be configured accordingly. Their various combinations result in different customized products according to customer requirement, technical requirement or according to the price of the final product. Every cell can be equipped with necessary components such as electrical, pneumatic, communication elements and with the software architecture in order to assure the modularity concept explained above.

In this research, bilevel modularity is introduced for the microfactory concept; modularity of the module itself and module within itself which enables reconfiguration of miniaturized system for further gains in product diversity, space, cost, flexibility etc. The proposed Flexible Process Oriented Modules (POMs) facilitates mass customization of products within a microfactory setting.

The bilevel modularity concept of the proposed microfactory modules is shown in the following illustration (Figure 5-2). The module takes the unassembled parts as an input and the assembled product/sub-product is the output of the assembly module. POMs, each configured for a different type of process, can be cascaded in order to form a complete microfactory. In the illustration, the assembly module developed is used as an example to show the concept of the bilevel modularity concept. The module within itself consists of different sub components for the process that will be realized in it. The components of the assembly module are shown in the figure and the modularity concept is shown using these sub components.

First layer of modularity is achieved by using different process oriented modules and forming the layout according to the product to be produced in the microfactory. Figure 5-3 illustrates examples of alternative layouts formed using different process oriented modules. This level of modularity in the illustration holds for any different kind of POM configured for any kind of production process. In the illustration three different possible layouts, each rectangle representing the projection of a POM, are shown. The layouts can be changed according to the process flow of the production and the number of POMs necessary for the microfactory.

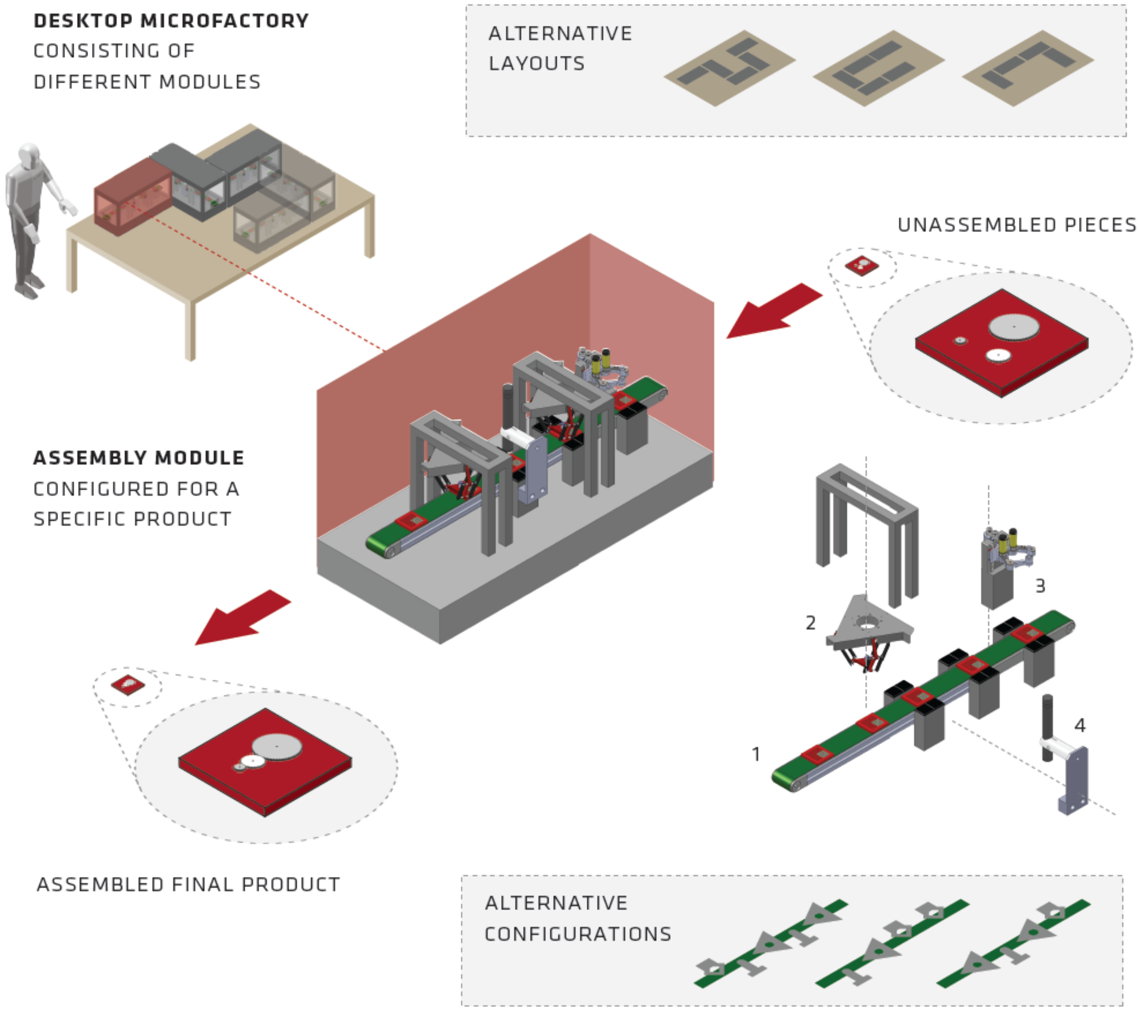


Figure 5-2 – Bilevel Modularity In Microfactories

Second layer of modularity is achieved within the module with the modular design and placement of the components that each module necessitates for the process to be realized within itself. Alternative layouts are illustrated in Figure 5-4 showing different placement of the main components of the assembly module. In the illustration three possible configurations of the assembly module components; Delta robot, pantograph and the vision sensor located one after another on the carriage unit, are shown. These components are designed in a modular way so that each unit can be located at any location within the modular cell at any number limited with the length of the carriage unit.

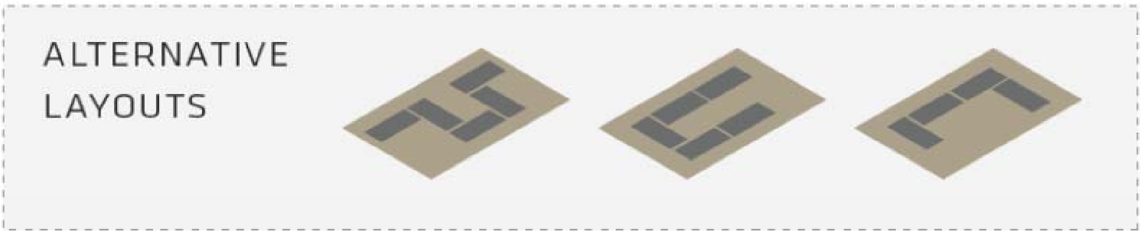


Figure 5-3 – First Level of Modularity

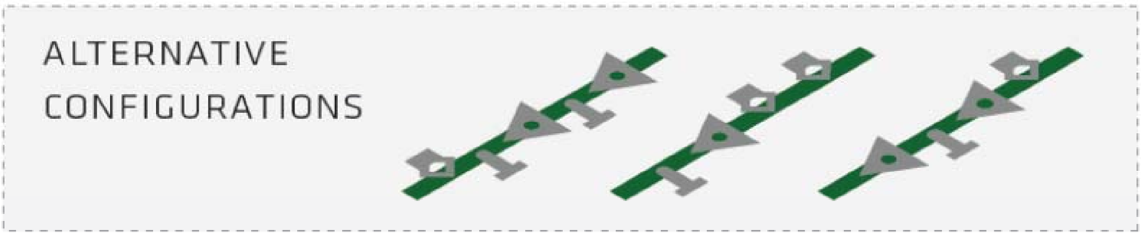


Figure 5-4 – Second Level of Modularity

The robotic assembly module realized as a POM explained above is shown in Figure 5-5. The system shown includes a vision sensor, a carriage unit, two Delta robots and controller hardware underneath the system. The carriage unit is equipped with necessary sensors in order to detect the trays that carry the parts to be assembled and stoppers at each station to position the trays at desired positions. The parts to be assembled moves on a tray carried by the conveyor in between stations and each station performs the task assigned for the formation of the final product outcome.

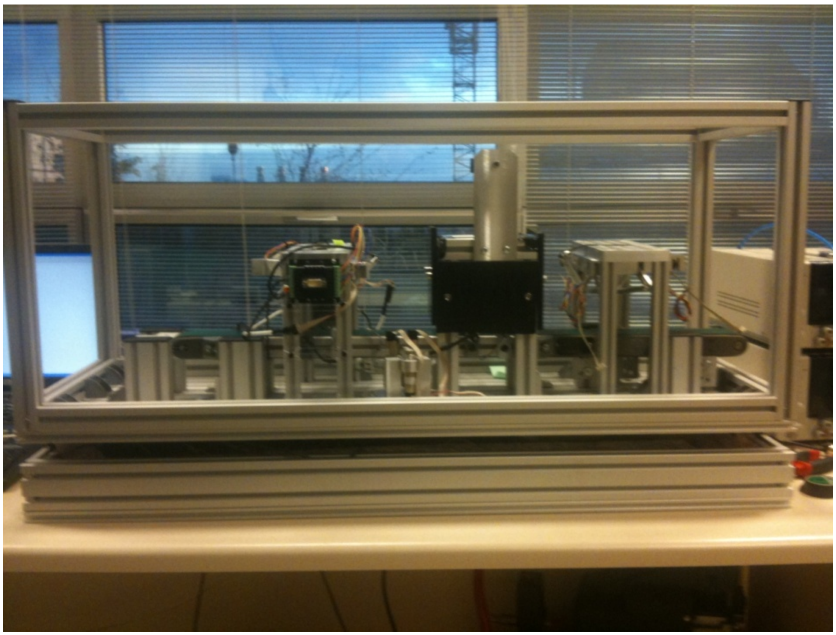


Figure 5-5 – The Assembly Module

5.2 System Supervision

In this section, supervision of the microfactory modules by means of software and electronics is explained. Software architecture is given with the brief explanation of software framework developed within the content of the ongoing PhD work [38] and the application of the framework to the microfactory module with the explanation of the system software is given. Additionally, the control system hardware developed for the modular structure of the system is explained.

5.2.1 Software Architecture

The modular structure of the microfactory necessitates modular and flexible software architecture for easy modification and reconfiguration of the system. A software framework which is reconfigurable with the ability to work on different platforms is developed within the context of another PhD work [38] in the Microsystems Laboratory. Other than the microfactory modules, the framework is used and tested on different platforms such as microassembly workstation, laser micromachining workstation, etc. The common point of all those systems is they are multi-degree of systems which involve high precision control and complex system supervision. The software is structured in order to handle these challenging issues with necessary functional layers. The software is firstly developed in the early research for the microassembly workstation which is explained in the following sections. The modular and flexible structure of the framework enables us to use it for the software structure of the microfactory module.

The layers of the framework are shown in Figure 5-6. The framework consists of two different components; one for the realtime modules for precise motion and process control and the other for non-realtime modules for offline task processing; GUI and MMI. The structure and modules of the software framework are defined in the following parts using the microfactory assembly module as an example.

The realtime part has two main layers; Motion and Process Control Layers. Both layers are sharing the hardware and communication interface layers. Different systems use different platforms for the motion and process control purposes. These platforms include both electronics hardware and the operating system (OS) running on the platform. The hardware interface layer standardizes the interface for the software-hardware interaction by a wrapper template for the electronics hardware utilizing the necessary functions for the major electronic interfaces (I/O functions).

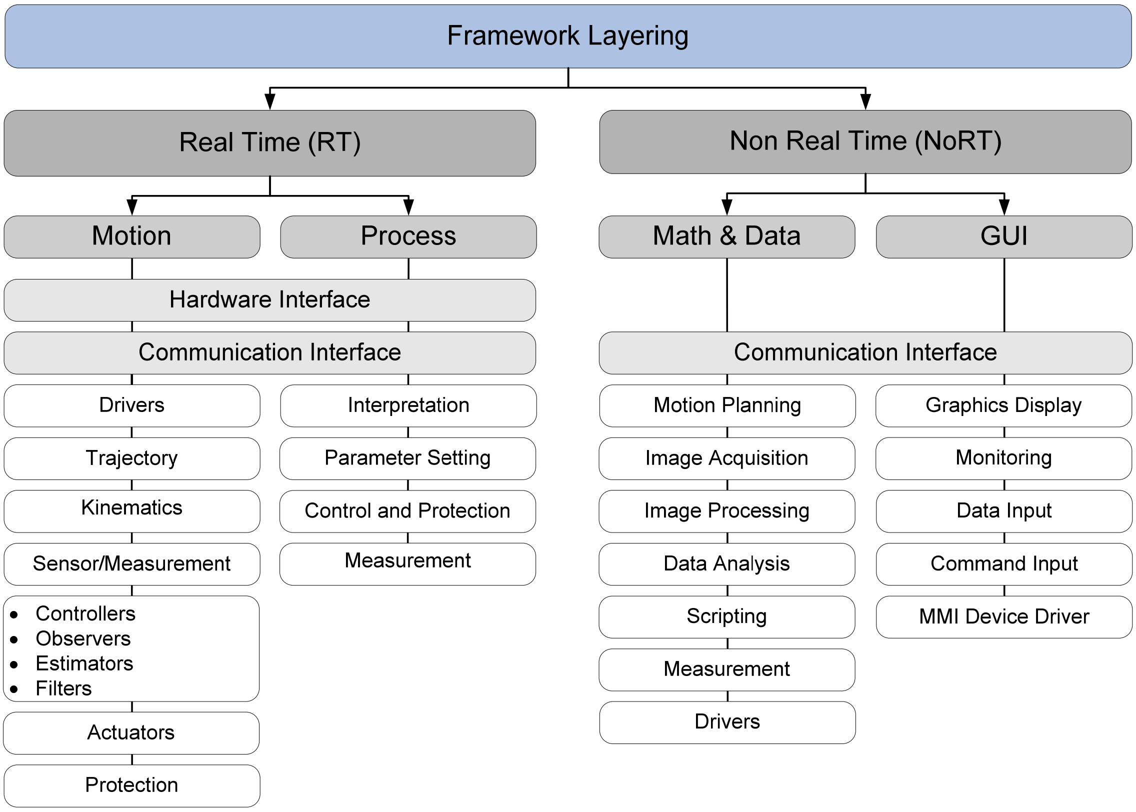


Figure 5-6 – Framework Structure [38]

The RT part and the NoRT part of the software for a system may run on different platforms/computers. The communication interface layer enables the data communication through several communication protocols in between the RT and the NoRT part of the software. This layer involves software functions generating the data packaging, sending, receiving and interpreting type functions according to the communication protocol available such as RS232, Ethernet, etc. For example; the microfactory assembly module is built onto two different platforms for experimental and performance purposes. Both systems are explained in further sections. The modular system is built upon a FPGA based RT Control computer enhanced with necessary modules burnt into the FPGA. The motion and process control layers of the software are running on the FPGA. Besides the hardware part written in VHDL and burnt into the system, the RT software runs on the embedded IBM PowerPC 405 processor inside the FPGA. The PowerPC exists as a hardware unit inside the FPGA. It does not consume any of the resources present in FPGA, but it facilitates implementation for some modules. The GUI and supervision software (NoRT part of the software) is written in C# and running on a PC. The communication between the RT computer and the NoRT computer is provided over RS232 protocol.

The motion layer has the modules as software functions to generate the desired motion by means of using different combinations of the motion layer functions in the system. These functions include every simple building block necessary for the construction of a motion control system. These function layers are;

- *Drivers* – Some of the actuators used to build up a motion system have their own electronics that must be used to enable motion which may need a software module for communication and input/output purposes in between the system software and the electronics. This layer of software module contains the software drivers in order to provide the data extraction to be used in the main software from the special electronics of the actuators.
- *Trajectory* – Functions that provide trajectory generation algorithms for single DOF actuators/stages or multiple DOF manipulators. Trajectory generation up to three DOFs is generated for the microfactory submodules.
- *Kinematics* – Functions of forward and inverse kinematics of manipulators for the transition between the task space and the joint space. The manipulators used in the system are two different parallel robots; Delta robot and pantograph. The analytical forward and inverse kinematics of both robots is implemented as functions in the control computer part.
- *Sensor/Measurement* – Any sensor readings and measurements for the motion of the actuators as feedback are provided by this layer of functions. Encoders, hall-effect sensors, laser sensor measurement, etc. can be given as examples. In the assembly module, for the motion part, quadrature encoders for the dc motors used in the manipulators, inductive sensors for conveyor motion, capacitive sensors for the additional piezo stages, etc.
- *Controllers* – Motion control methods are the most important module in order to achieve high precision motion. This layer of functions contains different motion control methods to be used according to the system needs like simple PID controller, sliding mode controller (SMC), adaptive control, optimal control, force control, passivity based control, non-linear control, etc. For the motion in the microfactory, PID and SMC based

controllers are mainly used for operational purposes. Other methods are available as function blocks and can be used for the experimentation with the manipulators.

- *Observers* – Observers can be defined as algorithms that combine the knowledge of the system with the sensor outputs in order to provide better results when compared to the traditional structures which wholly rely on sensors. The system input/output is typically combined with a mathematical model to predict the behavior of the system. A disturbance observer is the structure that estimates the disturbance acting on the system. In a motion control system, disturbance can be thought as the total load acting on the motion system plus the effect of changes in the system parameters.
- *Filters* – Filters are mathematical operations performed on a digital signal in order to reduce or enhance some aspects of the signal according to the requirements. LP filters can be given as an example that is used to filter the sensor outputs.
- *Actuators* - An actuator is a mechanical device for moving or controlling a mechanism or system. It is operated by a source of energy, usually in the form of an electric current, hydraulic fluid pressure or pneumatic pressure, and converts that energy into some kind of motion. The layer of functions for the actuator enables the input for the actuator to provide motion. Types of actuators used in the assembly module are dc motors, brushless dc motors, piezo actuators and a functional block exists for each of them in the software.
- *Protection* – The protection layer provides the security measures for preventing the system from any kind of damage. These functions in that layer are the limitations for the excessive motion, source energy input limitations like voltage and current saturation levels, etc. Any software function that helps to protect the system from possible errors or the environment with or without the inputs from the sensorial inputs is a part of that layer. The limitation of the robotic manipulators, energy limitations for the actuators like saturation voltages for protection and control purposes exists in this layer of the software.

The process layer of the RT part is defined to involve the necessary layer of functions to control processes other than motion. The functions in this layer allow the control of the processes that are needed in a system. These processes involve heating, curing, machining, cleaning, injecting, etc. which requires a device to be controlled for the effective functionality of the process.

- *Interpretation* – Some devices need special software for the operation and between that software and the device. A software layer is necessary to interpret the data which is generated by the special software. The interpreted data then fed to the device to perform its duties. The G-Code interpretation for a CNC machine is an example for that layer of software functions.
- *Parameter Setting* – This level enables the necessary parameters setting for the device to satisfy the necessary operating conditions for the process. In the assembly process, there are certain parameters to be adjusted during the process like the vacuum and motion idle times in order to ensure the realization of the assembly operation.
- *Control and Protection* – This level of functions takes necessary measures to control and protect the system from any possible damage during the operation.
- *Measurement* – The measurement layer is also necessary for the feedback from the different sensors or measurement units used in the device to perform an effective process.

The NoRT part of the software is running on a PC and implemented in C# for the assembly module. The NoRT software has two main layers; Math&Data and the GUI part. The Math&Data layer is dedicated to the mathematical operations and algorithms necessary for the implementation of the processes and supervision.

- *Motion Planning* – Motion planning algorithms are implemented offline in order to plan the motion for optimization and obstacle avoidance purposes. Mostly in the assembly processes these algorithms plays an important role for the efficiency and the reliable realization of the assembly process.
- *Image Acquisition* – A vision system may be necessary according to the type of the process in a system. Visual feedback gained great importance and now widely used in industrial systems. Several vision sensors with different communication protocols are available in the market and image acquisition is of great importance in image based systems. Most common types of the cameras used are IEEE 1394 Firewire, USB or Ethernet cameras. This layer has the necessary functional modules for the acquisition of the image through these communication protocols. In the assembly module, a microscope with a USB camera is used as the vision sensor and the image acquisition is realized over USB using the functions defined in this layer.
- *Image Processing* – This layer involves the necessary image processing methods for the vision system to perform its assigned duties in order to provide visual feedback

for the system. A vision interface for the microfactory assembly has great importance for the determination of the position and orientation of the parts to be assembled with respect to each other and the manipulation tool. The full automation of the assembly process can be achieved by these means. The effectiveness of a vision system lies on the accuracy of the visual feedback it provides such as recognizing the geometries and position of 3D microparts, path generation for the manipulators and providing the necessary position feedback for an assembly process. The data necessary for the motion can be achieved from the visual feedback by using the necessary image processing algorithms. Image processing methods that are used in the module are explained in the following sections.

- *Data Analysis* – In a production unit or system, for the optimization of the process there are measures showing the efficiency of the system. The production rate, quality are some of the factors representing the efficiency. The data extraction from the system can be maintained by necessary sensors or devices and this layer provides functions that transform the raw data to meaningful measures or graphs showing the performance of the system.
- *Scripting* – Scripting layer brings versatility to the system by means of enabling writing scripts of different complexity levels by different users in order to define upper level functions for the specific systems. With scripting the user can write C# code using all of the functions developed for the system including all the functions for the peripherals and move commands. The system combines the script code with the available libraries for the peripherals and the system and then the code is compiled and finally it is executed to run the script during runtime.
- *Measurement* – Any type of measurement (hardware or software) that is not subject to real time processes are defined in this layer.

The GUI is the visual software that enables the interaction between the operator and the system. It consists of main functional blocks that allow the operator to observe and intervene the features of the system using the graphical or numeric features and command input blocks that are presented on the GUI.

- *Graphics Display* – The GUI may involve some graphical features to show the operation, performance, indicators related to the process, guidelines, etc. to provide necessary data or cautions about the system. These may be alarm signals indicating the errors occurred on the system or highlighted features to point out the important aspects that must be taken care of for the planning and control of the operation. In the GUI of the module, positions of the parts to be manipulated are displayed on grid structure for the operator to define the assembly task.

- *Monitoring* – In this layer there are functional modules that indicate the states of the overall system. The visual information captured from the vision system, any measured data like the current positions of the manipulators, data related to the production or any other feature necessary to be monitored are displayed on the GUI by the functional blocks provided by this layer of the software.
- *Data Input* – The operator can intervene the system operations using numeric data inputs. These data may involve some parameters for the functional operation of the devices, data related to the production process, position reference for the manipulators or any other numerical data input related to the system operations.
- *Command Input* – Command inputs are predefined structures for the control or operation of the system. This layer contains a command list that is interpreted and then sent to the system to realize a desired action. GO-MOVE-START-STOP type operations that involve a sequence of actions are defined as a script and using the GUI blocks defined as text input or button, the commands can be given to the system to be processed. There are special buttons on the GUI defined as a command input to the microfactory module.
- *MMI Device Driver* - The operator interference to the system can be realized by means of a MMI Device where GUI is the guidance of the operator. Any haptic device, mouse, operational panel can be installed and used for system interaction by the operator. And for the functional operation of these devices a driver is necessary to get and transform the input into meaningful commands for the system.

5.2.2 Software Implementation

The implementation of the software is realized using the software framework defined in the previous section. The basic modules of the system are used to generate functional upper blocks representing each modular unit of the microfactory. The content and structure of the software is shown in Figure 5-7. The RT and NoRT parts and the contents defined as layers which are used in the microfactory module are shown with respect to the framework structure.

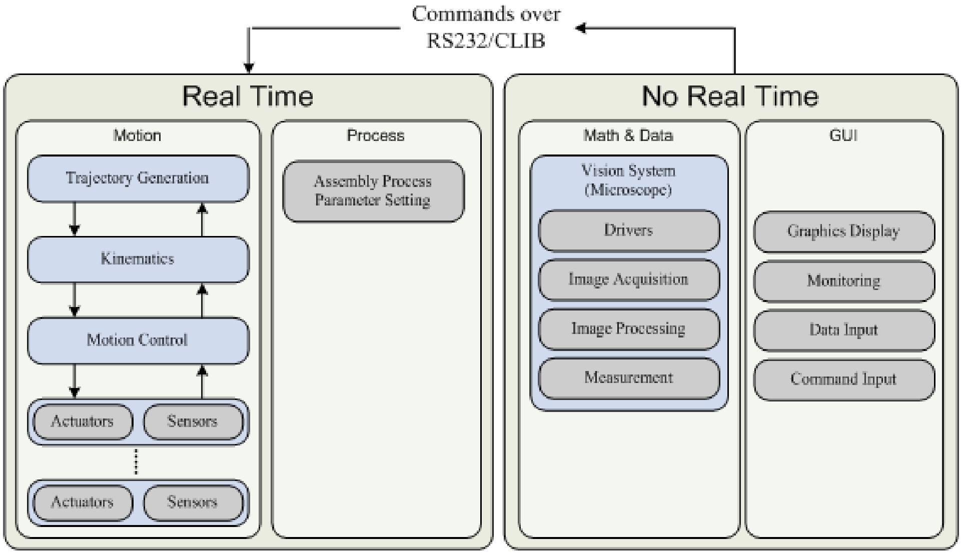


Figure 5-7 – Software Structure for Microfactory

The implementation of the software structure is given for the Delta robot as an example in Figure 5-8. The position reference is retrieved from the GUI with the user input during task generation. The communication layer transfers this position data from the NoRT layer to the RT layer. Given the reference value for the robot and the measured position of the robot, the trajectory generation layer calculates the necessary input position values for each coordinate. The input values for the task space are then converted to joint space using inverse kinematics and then the control is applied for each axis of the robot. The protection layer puts necessary limitations according to the type of the actuator and filtered out control inputs that are fed to the actuators.

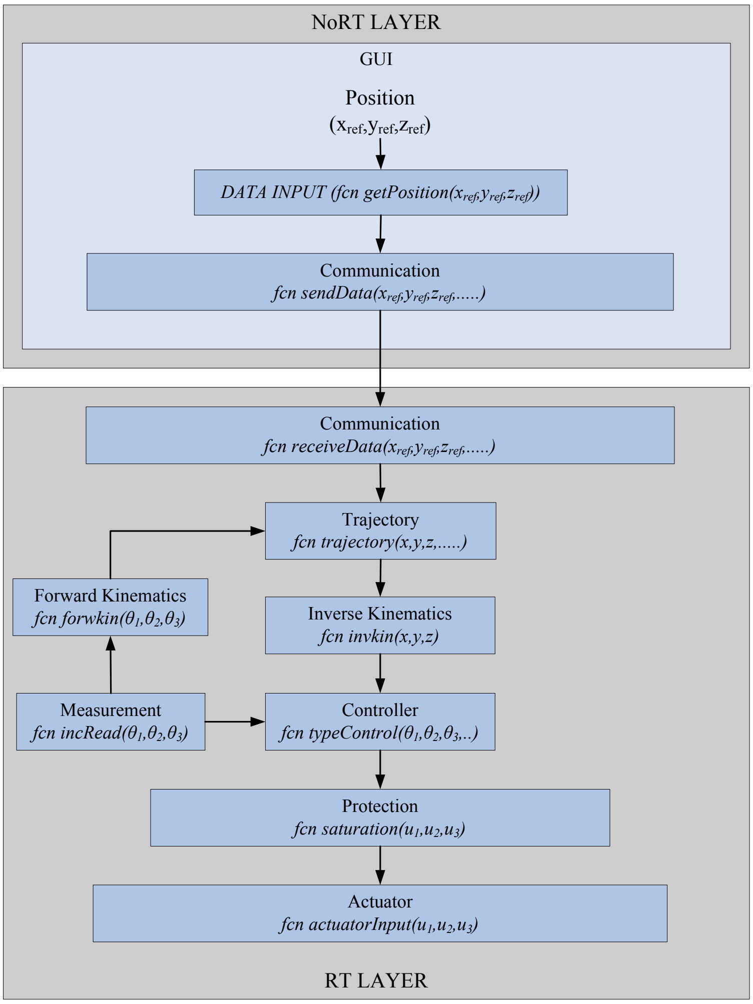
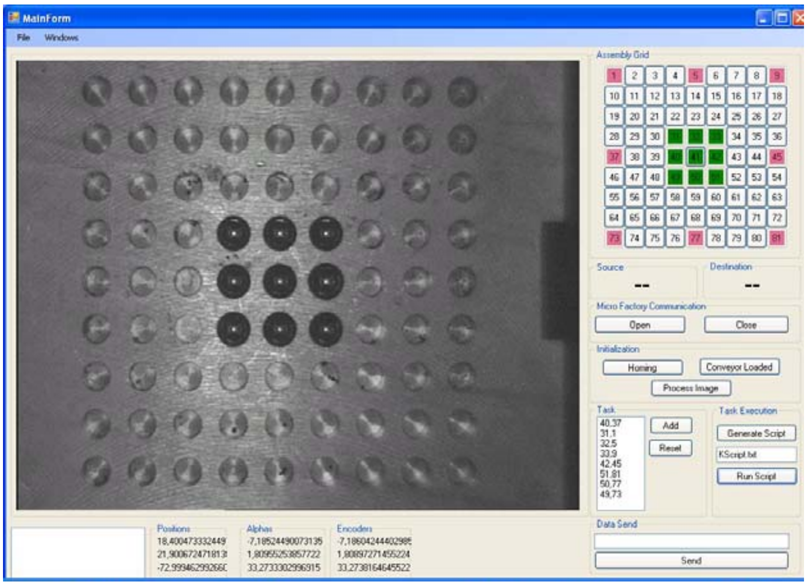

NoRT LAYER
GUI
Position
$(x_{ref}, y_{ref}, z_{ref})$
DATA INPUT (fcn getPosition($x_{ref}, y_{ref}, z_{ref}$))
Communication
fcn sendData($x_{ref}, y_{ref}, z_{ref}$,.....)
Communication
fcn receiveData($x_{ref}, y_{ref}, z_{ref}$,.....)
Trajectory
fcn trajectory(x,y,z,.....)
Forward Kinematics
fcn forwkin($\theta_1, \theta_2, \theta_3$)
Inverse Kinematics
fcn invkin(x,y,z)
Measurement
fcn incRead($\theta_1, \theta_2, \theta_3$)
Controller
fcn typeControl($\theta_1, \theta_2, \theta_3$,..)
Protection
fcn saturation(u_1, u_2, u_3)
Actuator
fcn actuatorInput(u_1, u_2, u_3)
RT LAYER

Figure 5-8 – Delta Robot Software Structure

5.2.2.1 NoRT Software

The NoRT part of the software involves the GUI and Vision Software as the main units. In Figure 5-9, the GUI of the microfactory assembly module is shown. The GUI is generated for the experiments with steel spheres in order to test the performance of the system. It consists of the visual information display, task generation module, communication module, data monitoring module and command input module. It also includes some debugging features since it is used in the development phase of the microfactory concept.

The main supervision software is developed using C# language. The image captured by the camera attached to the microscope is displayed on the GUI for the visualization of the objects to the operator. Using the image processing methods, determined positions of the parts are displayed on the grid structured plate (graphics display) and the information is conveyed to the task generation module. In this module, the user can define a task plan to be realized by the manipulators in the system. The task plan can be reconfigured again from the GUI. The manipulator can also be controlled from the GUI using the numeric position reference blocks (data input). The assembly module includes Delta robots as the manipulators. The homing procedure can be initiated from the GUI (command input) and current positions of the Delta robot can be observed (monitoring) and position reference input can also be given (data input). The assembly process can be controlled step by step with the monitoring units and the command input units using the GUI.



1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

Source: --

Destination: --

Task:
40,37
31,1
32,5
33,9
42,45
51,81
50,77
49,73

KScript.txt

Positions	Alphas	Encoders
18,400473332449	-7,18524490073135	-7,1860424440298
21,900672471813	1,80955253857722	1,80897271455224
-72,999462992660	33,2733302996915	33,2738164645522

Figure 5-9 – Assembly Module Test GUI

The communication between the GUI and the real time computer is provided over RS232 in the FPGA system. The dSpace cards are mounted over ISA Bus to the same computer with the NoRT part of the software and the communication between the NoRT and the RT part is provided by a special software library called CLIB.

5.2.2.1.1 Vision System Software

The vision system software for the assembly module is configured to detect the positions and orientations of the parts to be manipulated. The position and orientation values for the parts are fed to the task generation screen of the GUI and then the operator defines the assembly task. The position values detected on the image (image coordinates) are then converted to the world coordinates with the calibration between the image and world space. The world coordinates are then sent to the manipulator as position reference for the pick place operations.

Image acquisition and image processing software is written using MVTec Halcon 9.0. The software generated by this platform is then converted to C# and then embedded into the main system software as image modules.

The first step of any vision system software is the image acquisition. The image must be acquired adequately so that the further image processing methods can be applied. The level of the loss of information from the image should be low in order to extract the necessary features from the image. The image acquisition needs necessary drivers for the cameras used in the system since they have different communication protocols like IEEE1394, USB, Ethernet, etc. After the image has been acquired, various methods of image processing can be applied to the image to extract necessary information.

Camera calibration is defined as the determination of the necessary parameters for the mapping between the world and image coordinates. For the modeling of the optical projection of the world coordinates, $P_w = (x_w, y_w, z_w)$, into image coordinates, (r, c), first the transformation of the world coordinates into camera coordinates, $P_c = (x_c, y_c, z_c)$ is realized and defined as;

$$\begin{pmatrix} P_c \\ 1 \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P_w \\ 1 \end{pmatrix} \tag{5 - 1}$$

where R is $3x3$ rotation matrix and T is the translation defining the position of the camera in the world coordinates system.

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}, T = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{5 - 2}$$

The projection of the point $P_c = (x_c, y_c, z_c)$ in camera coordinates into the image plane coordinates (u, v) is defined as;

$$\begin{aligned} u &= f\,\frac{x}{z} \\ v &= f\,\frac{y}{z} \end{aligned} \tag{5 - 3}$$

where f is the focal length of the camera.

The structure of the camera lenses causes radial distortions in the image which can be modeled with the division model [39] using only one parameter, κ, to model the distortions and the new corrected image plane coordinates, (u', v'), are expressed as;

$$u' = \frac{2u}{1 + \sqrt{1 - 4\kappa(u^2 + v^2)}}$$
$$v' = \frac{2v}{1 + \sqrt{1 - 4\kappa(u^2 + v^2)}} \tag{5-4}$$

The transformation of the image plane coordinates, (u, v), into the image coordinates, (r, c), is derived as;

$$c = \frac{u'}{S_x} + C_x, \qquad r = \frac{v'}{S_y} + C_y \tag{5-5}$$

where S_x and S_y represent the horizontal and vertical distance between the two neighbor cells on the image sensor, C_x and C_y represent the image center point row and column coordinates.

The camera calibration process determines the external parameters, R and T, and the internal parameters, $f, \kappa, S_x, S_y, C_x, C_y$ of the system from a set world coordinates and the corresponding image coordinates. The determination is realized with the minimization of the sum of the squared distance between the projection of the world coordinates in the image and the corresponding image coordinates. The convergence of the minimization provides the determination of the internal and external camera parameters. The initial values of these parameters are provided by the manufacturer and play an important role for the minimization process.

For the calibration of the system a custom calibration plate matching the size of the visible workspace of the camera has been designed. A known pattern used by the Halcon platform has been printed on a paper that matches the size of the trays used in the system. Given some initial values provided by the camera manufacturer for the camera parameters, the 3D locations of the circular calibration marks on the plate can be projected into the camera plane. Then, the determination of the camera parameters are realized such that the minimization of the distance of the projections of the calibration marks and the mark locations extracted from the image is achieved. This minimization process returns fairly accurate values for the camera parameters. In order to obtain the camera parameters with the highest accuracy, several images of the calibration plate is taken where the plate is placed and rotated differently in each image to use all degrees of freedom of the exterior orientation.

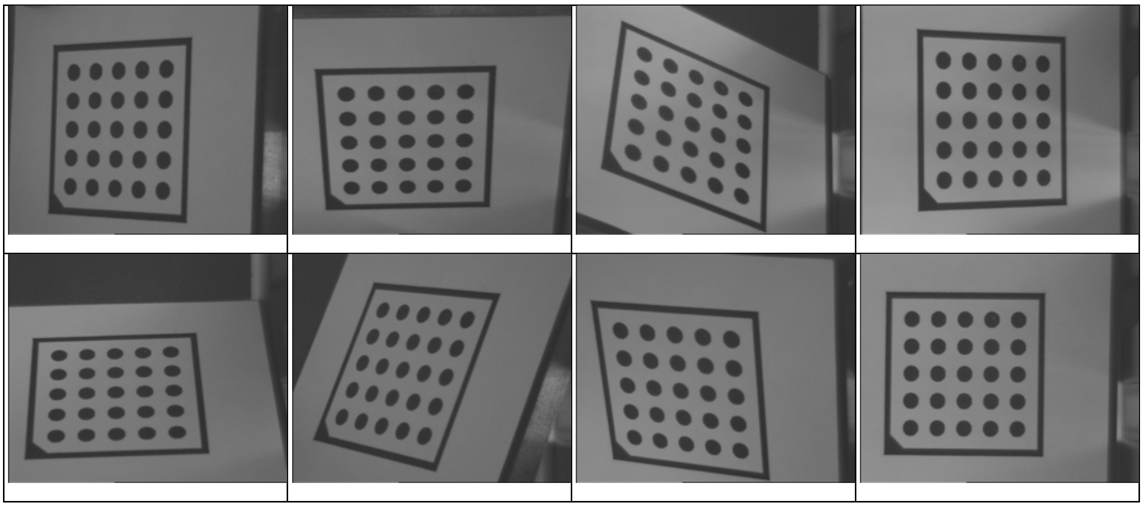


Figure 5-10 – Several Images of the calibration plate with different orientations

The radial distortion is compensated using the external and internal parameters determined by the calibration process. The original image captured and the rectified image with the radial distortion removed are shown in Figure 5-11.

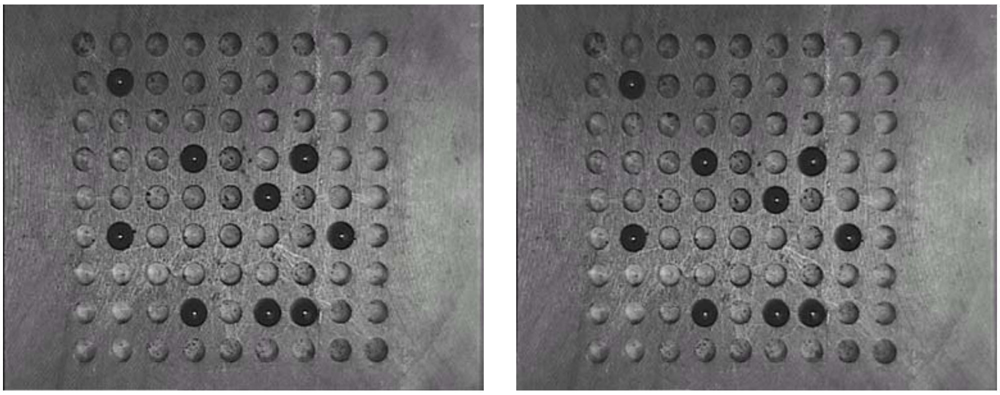


Figure 5-11 – (a) Original Image (b) Rectified Image

For the determination of position and orientation of parts to be manipulated for the assembly process, it is necessary to distinguish some meaningful data to identify the objects which are called image features. Feature extraction can be defined as a special form of reduction in dimensionality and one of the main issues in computer vision since the extracted features must be meaningful in the sense that they must be sufficient enough to describe the model and properties of the desired portions or shapes in the image.

In order to reduce the computational power necessary for the image processing ROI (Region of Interest) selection plays a key role. For that reason, the ROI on the tray image is selected and that region is extracted for the application of further image processing techniques. The rest of the techniques are implemented on this part of the image. In Figure 5-12(a), the ROI is shown with a rectangle and the extracted image is shown in Figure 5-12(b).

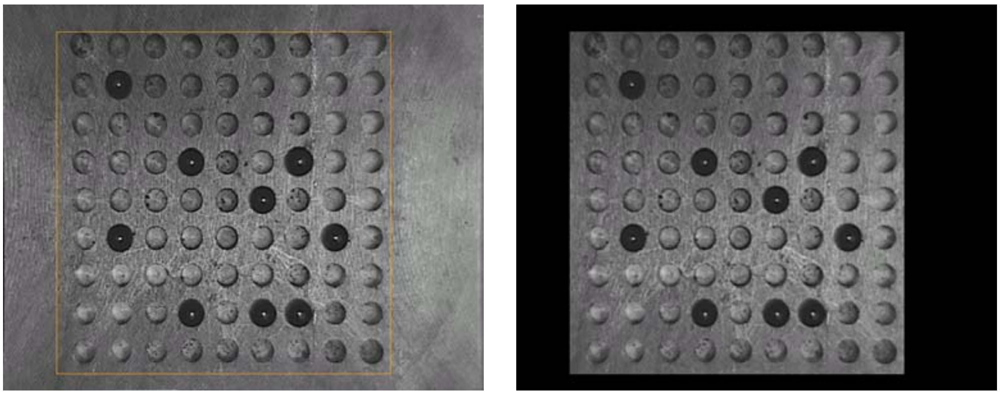


Figure 5-12 – (a) POI Selection (b) POI Extracted

The average (mean) filter is applied to the image for smoothing and elimination of the noise. Let S_{xy} represent a set of coordinates in a rectangular subimage window of size mxn, centered at point (x,y). The arithmetic mean filter computes the average value of the initial image, $I(x,y)$ in the area defined by S_{xy} and the value of the restored image $I_R(x,y)$ at any point in the image (x,y) is the arithmetic mean computed with the pixels in the region S_{xy}.

$$I_R(x,y) = \frac{1}{mn} \sum_{(s,t)\epsilon S_{xy}} I(s,t) \tag{5-6}$$

The operation is realized using a $3x3$ convolution mask with the coefficient value of $1/mn$. The resulted image is shown in Figure 5-13(a).

A segmentation process is necessary for the separation of the regions in the image that corresponds to the interested objects from the background regions. The simple way to do that is based on the different intensities in the back and foreground of the image. It is the action of transforming a grayscale image into a binary image based on a threshold value, T, determined according to the intensity values. After thresholding, the black pixels correspond to the background and the white pixels correspond to the foreground of the image or vice versa. The threshold value selected for the process represents the boundary between the intensity of the back and foreground regions of the image. During the process, intensity value of each pixel is compared to the threshold value and then the value of the each pixel is set to white or black.

$$g(x, y) = \begin{cases} 1 & if\ f(x, y) > T \\ 0 & if\ f(x, y) \leq T \end{cases} \quad (5\text{-}7)$$

where $g(x, y)$ represents the thresholded image and $f(x, y)$ is the gray level image. The histogram of an image which shows the occurrences of the each gray level on the image is helpful for the determination of the threshold value. The threshold image is shown in Figure 5-13(b).

For the determination of the parts in the image some morphological methods are applied in order to keep the features that are of interest for the image processing. These processes include two basic morphological operations in image processing; erosion and dilation. The other operations applied are the derivatives of these operations with different size of filters. The process is shown in Figure 5-14.

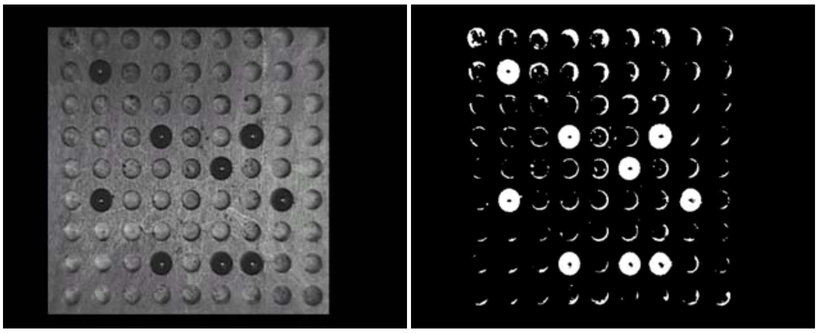
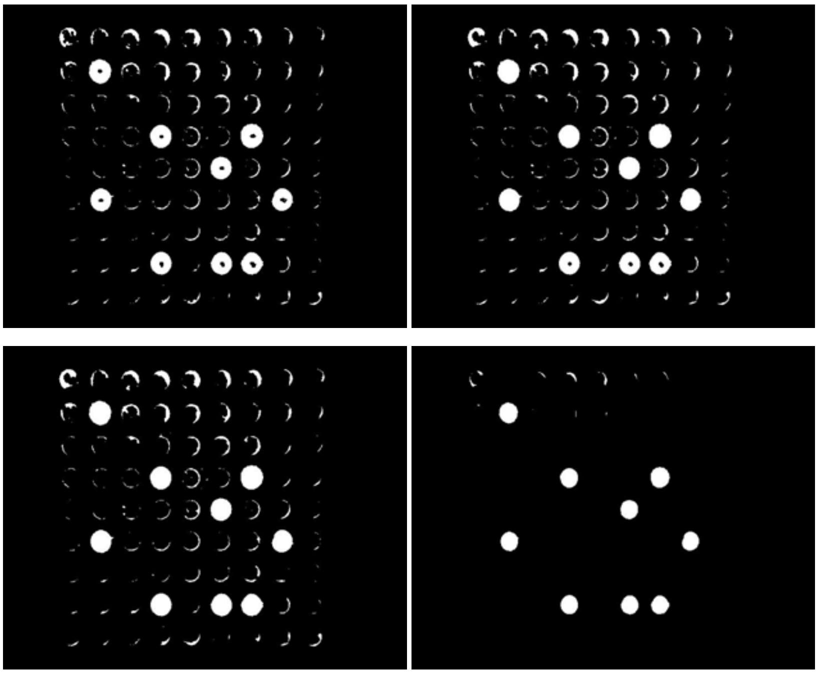


Figure 5-13 – (a) Image Mean of POI (b) Threshold Image



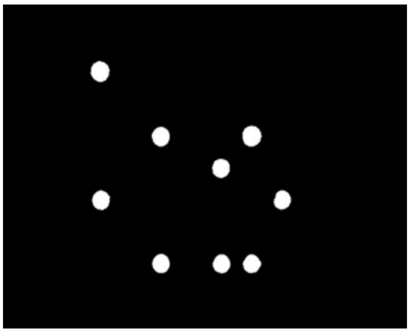


Figure 5-14 – (a) Eroded Image (b) Closed Image (c) Fill up (d) Eroded Image (e) Selected Regions

The objects of interest in the image are determined with the image processing methods and the positions of the objects in the image are determined according to the center of area of the objects in the image. The number of the objects determined in the image is displayed and the locations of the objects are marked and shown in the image. The resulting image is shown in Figure 5-15.

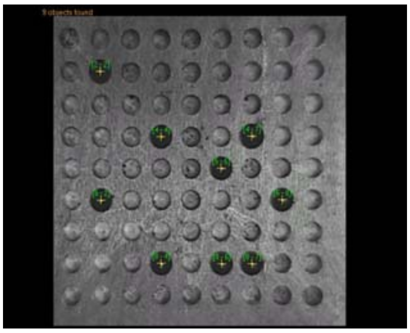


Figure 5-15 – Object Positions Displayed on Image

The image processing methods are firstly generated for the position determination of the spheres which are used for the pick place operations. Then the algorithms are modified for the position determination of different objects. The resulting images for the position determination of a 2mm nut and SMD (Surface Mount Devices) Resistors are shown in Figure 5-16.

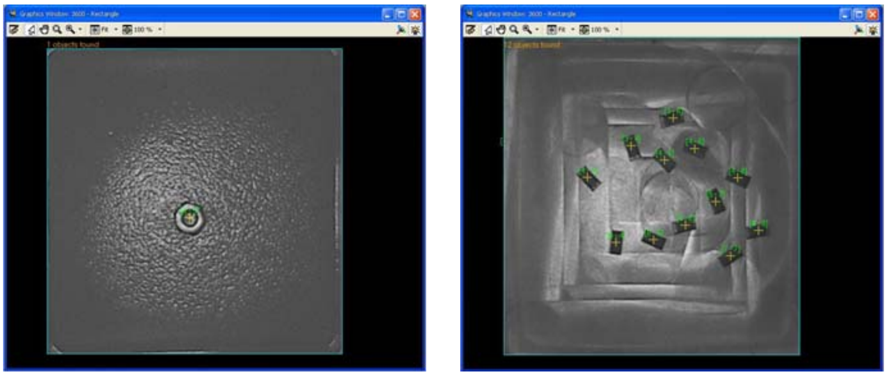

Graphics Window: 3600 - Rectangle
1 objects found
Graphics Window: 3600 - Rectangle
12 objects found

Figure 5-16 – (a) 2mm nut (b) SMD Resistors

5.2.2.2 The RT Software

5.2.2.2.1 FPGA

The RT software for the microfactory module is implemented on two different platforms. The first one is FPGA and the second one is the dSpace platform. FPGA is a field programmable gate array in which gates are reconfigurable switches for interconnections. It can be programmed using hardware description languages (HDL), which results in the implementation of the algorithms physically in the integrated circuit (IC).

A PC configured as the main supervision computer, FPGA is configured as the control computer and the modules in the control computer are generated within the FPGA using the hardware description languages and some modules of the system have been implemented in software running on the embedded IBM PowerPC 405 processor inside the FPGA.

The physical interfacing parts have been implemented as hardware in Verilog HDL by using the Xilinx ISE tool. The software framework is implemented in the software part implemented as the control library on the processor. The physical interface blocks of the control library consist of encoder blocks and different types of pulse width modulation (PWM) generators.

The quadrature encoder interface requires the inverses of the encoder signals as inputs which increases the precision of the encoder block. If the encoders of the actuators do not provide the inverse of the signals, the interface block can also be implemented as a dual encoder hardware block. The output of the encoder block is implemented as a 32-bit counter and the sequence of the block is shown in Figure 5-17.

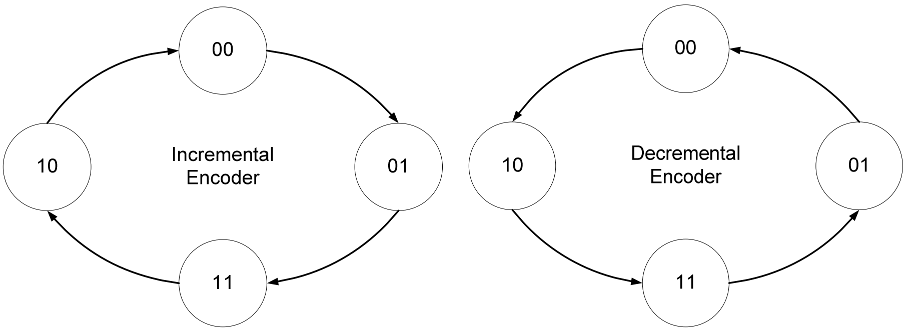


Figure 5-17 – State Sequences for the encoder block

For homing purposes, a home input is enabled in order to set the counter value to a constant. The home input can be triggered with integrated sensors to the system so that the encoder values can be set to a predefined value in order to adjust the position value of the actuators or the mechanisms.

Pulse width modulation (PWM) is the variance of the duty ratio of a square wave in one cycle. Frequency and the resolution related to the accuracy of the output signal are the two parameters to be considered for a PWM signal. Two types of PWM Generators are implemented in FPGA; one with the direction module and the other with the bias module. The PWM Generator with direction module takes a 32-bit integer value as input, and then the input gets through saturation where the threshold value is determined by the bit-width setting of the block. The direction signal is the most significant bit of the saturated signal. The rest of the input is sent to the comparator module where the output is logic-1 until the value is equal to the input. Then the output becomes logic-0. The structure of the PWM Generator with direction is depicted in Figure 5-18.

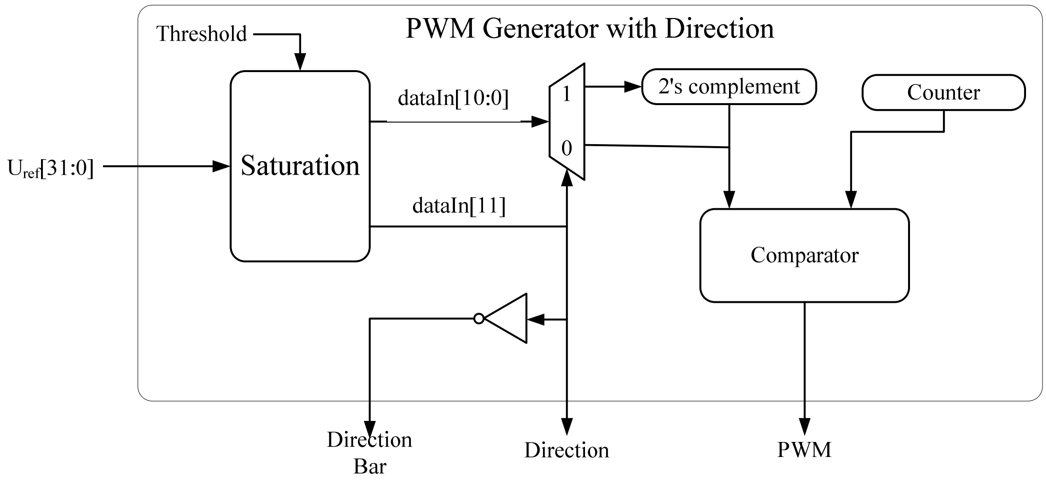


Figure 5-18 – The structure of the PWM block with direction signals

PWM Generator with bias module has the same structure with the PWM with direction module. The difference is that a bias value is added to the saturated signal in order to have %50 duty ratio at zero value input. The biased signal is sent to the comparator block as it is the case in the PWM with direction module. The structure is depicted in Figure 5-19.

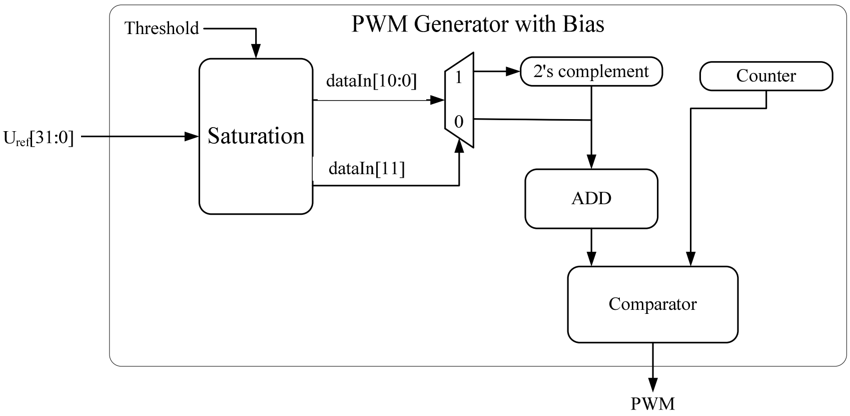


Figure 5-19 – The structure of the PWM block with bias

The real time loop is running on the embedded IBM PowerPC 405 processor inside the FPGA. The software functional modules defined in the previous sections in the RT part of the software are implemented on the processor using the C language. The functions are used in the real time loop for the control of the system and the communication between the RT part and the NoRT part is established using RS232 protocol.

5.2.2.2.2 DS1103 PPC Board

The DS1103 PPC board has its own connector panel and electronics interfaces. It also has its own software library for these interfaces. The PPC board is integrated to the main supervision computer from ISA bus on the motherboard. The real time software is directly transferred to the system since dSpace ControlDesk software also uses C language. For debugging purposes, at the first phases of the system development the GUI of the dSpace ControlDesk is used. The communication between the RT part and the NoRT part of the software is realized using a communication library (CLIB) provided by dSpace which is written in C language. CLIB provides communication over PCI bus, ISA bus and fiber optical network.

5.2.3 Electronics Design

The modularity feature of the module requires control hardware which is capable of handling different types of actuators since according to the needs of the process to be realized in the module, any actuator or mechanism composed of different actuators might be the matter of choice. For that reason the control electronics needs to be developed to satisfy the handling of different types of actuators. The control system should also have a modular structure and should be compact since the concept of process modules requires these properties. Field programmable gate arrays (FPGA) bring flexibility to the control electronics design with reduced production cost and lower implementation time. With the modular programming and creating basic blocks to be used for the control of the actuators within the factory module, it is considered as a good option for its compatibility for the modular system concept.

In order to provide the interface of different type of actuator inputs/outputs to the FPGA an interface board is designed supporting 12 encoder interfaces, 12 24V inputs and 12 24V outputs. The interface board is designed in such a way that different type of motors can be used in the system. For that purpose different type of driver modules are designed which can be easily attached to the main interface board. Different encoder interface cards are also designed since the encoder inputs can vary for different actuators. The control hardware is capable of managing 12 DOF with necessary inputs and outputs. The board that is used is Digilent Inc. Xilinx Virtex II Pro. The control electronics, the connector board and the driver modules are shown in Figure 5-20.

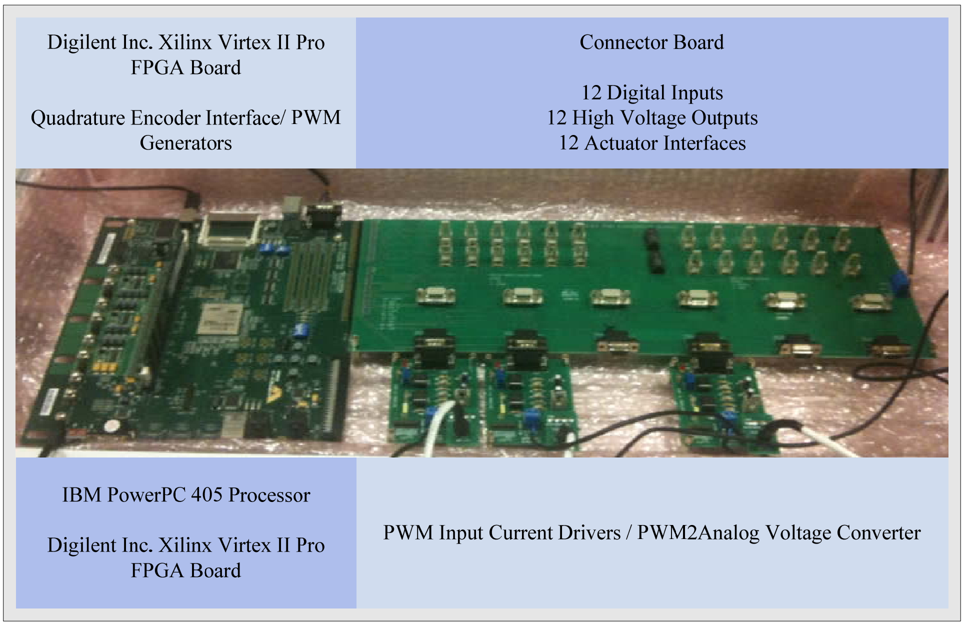

Digilent Inc. Xilinx Virtex II Pro
FPGA Board
Quadrature Encoder Interface/ PWM Generators
Connector Board
12 Digital Inputs
12 High Voltage Outputs
12 Actuator Interfaces
IBM PowerPC 405 Processor
Digilent Inc. Xilinx Virtex II Pro
FPGA Board
PWM Input Current Drivers / PWM2Analog Voltage Converter

Figure 5-20 – FPGA Electronics

The layout of the control electronics is designed to fit underneath the microfactory assembly module with the extensions of the modules of drivers, etc. The placement of the control electronics underneath the module is shown in Figure 5-21(a) and the PWM2Analog Voltage Converters integrated to the controller board in order to show the modularity of the electronics is shown in Figure 5-21(b).

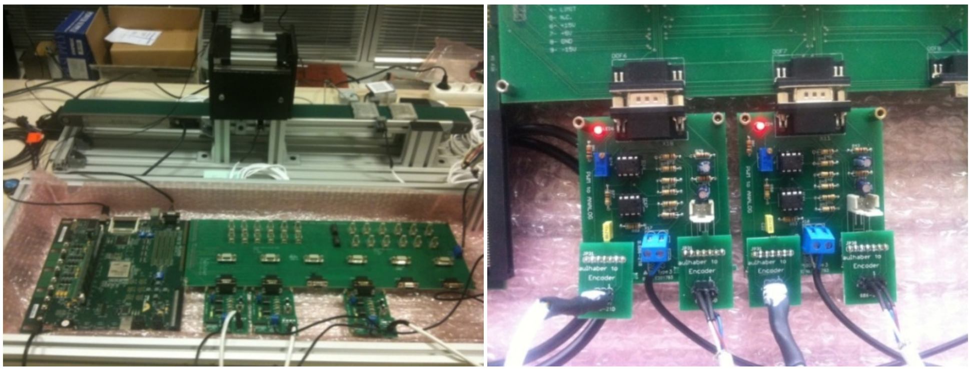


Figure 5-21 – (a) Assembly Module Control Electronics (b) PWM2Analog Converters/Drivers

Different types of actuators need to be driven by different type of drivers. As a necessity of the modularity, the control platform must be suitable for any type of actuator. The drivers of various types of actuators are commonly driven by an analog signal in ±10V range. The output of the FPGA system is achieved by PWM signal generators which are mentioned in the previous section so that a PWM-to-analog circuit has been designed to provide the input for the drivers. The input stage of the circuit consists of two cascaded passive low-pass filters. The second low-pass filter aims to diminish the ripple in the outcoming DC signal. The signal then passes through level shifter and gain stages in order to fit in the $\mp 10V$ range. The level shifter and the gain stages are designed with trimpots, which allow the adjusting of the level offset and the gain with high precision.

In order to drive the actuators which do not have its own electronics, a current driver integrated PWM to Analog Converter is also designed. The circuit has the PWM to Analog converter circuit as the first phase of the circuit and then the analog voltage is fed to the current driver which is designed as a high resolution driver with TDA2040 20W Hi-Fi Power Audio Amplifier as a classical op-amp mode to control the output current on a resistor by measuring the voltage on it.

Different actuator connector interface boards are designed for the interface of the actuators to the driver circuits. The connectors are designed for specific type of dc motors that are used in the assembly module. It has a standard pin header connector on the driver side and new designs can be easily made when different actuators are needed to be used in the system. It has the sensor measurement pins and the actuator input pins converted to the standard pin header configuration.

The main control architecture of the microfactory assembly module is shown in Figure 5-22. The system supervision computer communicating with the control computer over RS232 conveying the information in both ways. All the main components of the system are shown in interaction with each other.

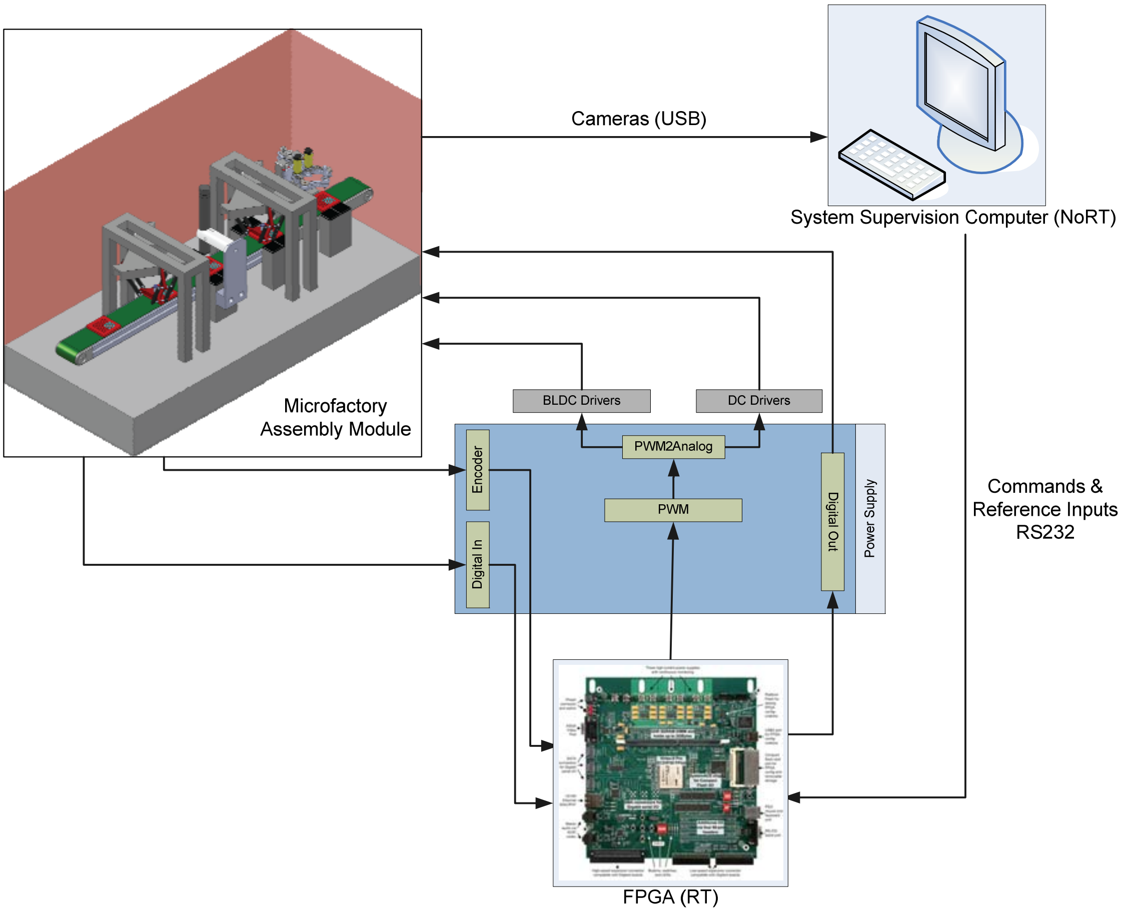


Figure 5-22 – Assembly Module Control Architecture with FPGA

As an alternative to the FPGA hardware, the control hardware is designed using DS1103 PPC Controller Board as the system control computer. The software for the ControlDesk software environment of the dSpace uses the C language so that the software structure is directly transferred to this system. The controller board has 8 encoder interfaces, 50 bit-I/O channels, 36 A/D channels, and 8 D/A channels which satisfy the needs for the assembly module. The DS1103 PPC board and the system computer with ControlDesk GUI are shown in Figure 5-23(a) and Figure 5-23(b).

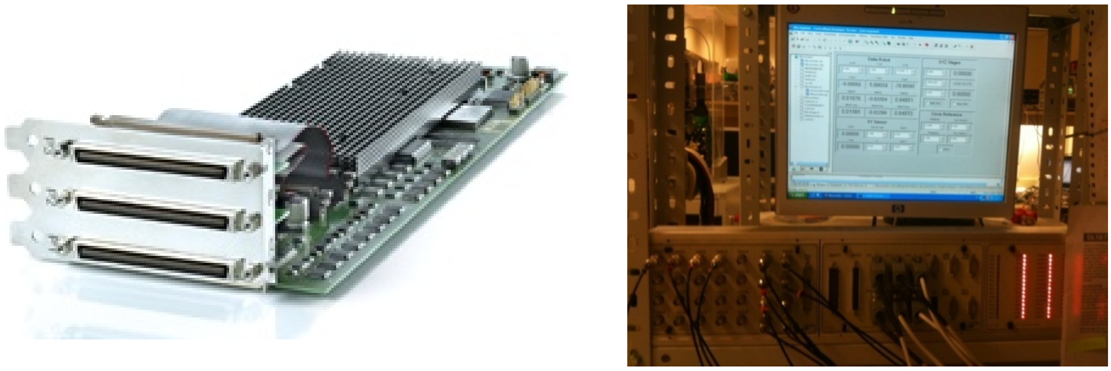


Figure 5-23 – (a) dSpace DS1103 Controller Board (b) System Control Computer and Interface

The control architecture of the microfactory assembly module with dSpace configured as the system control computer is shown in Figure 5-24. Necessary electronics are developed as drivers for the actuators which are noted in the system architecture shown in the figure.

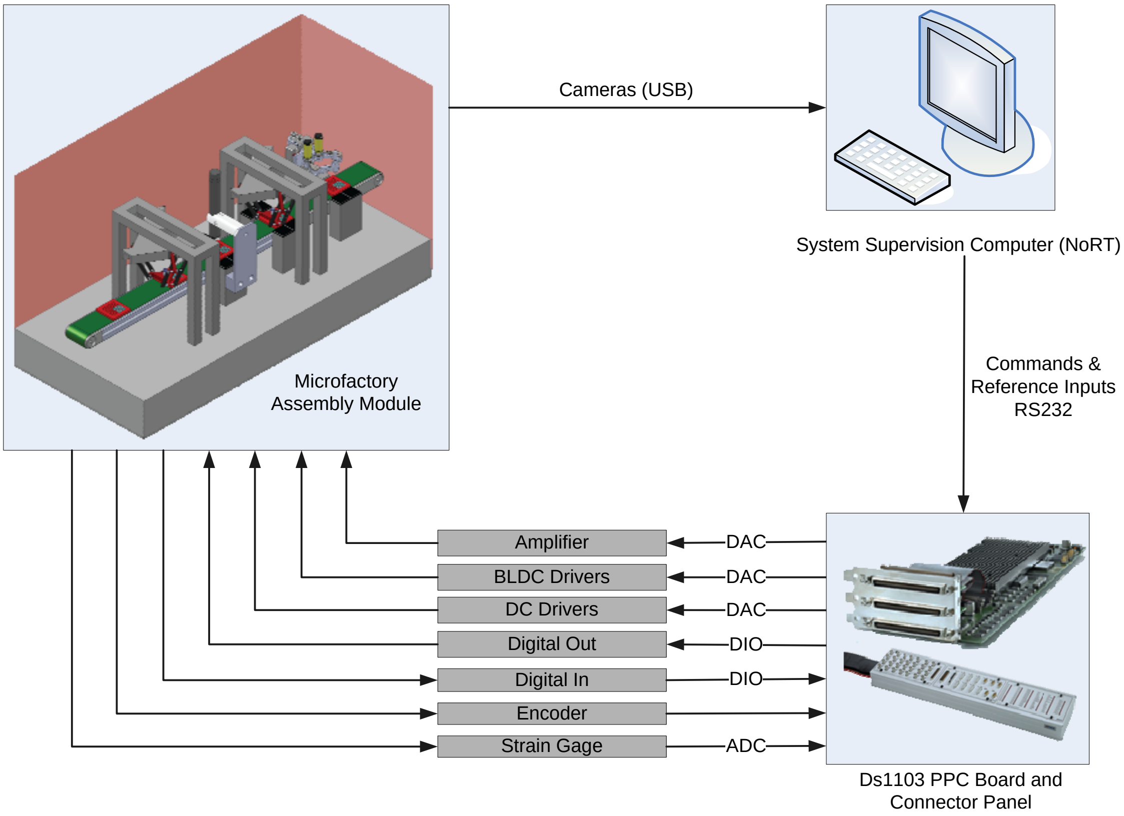


Figure 5-24 – Assembly Module Control Architecture with DS1103 PPC Board

6 ASSEMBLY MODULE

Within the concept of microfactory POM design, an assembly module is developed in order realize the proof of concept. The assembly module is designed and developed to be a whole assembly process module which may consist of several manipulators and other task units according to the needs of the process to be implemented. For the assembly module, adaptation of the previously designed microassembly unit is considered and the microassembly manipulator system is explained in the following sections. Several manipulators for the realization of high precision manipulation tasks are also developed. The assembly module is a combination of several modules that can realize the defined tasks in order to implement the whole process defined for the module. The software and electronics of the module are explained in the previous sections. The following sections are dedicated to the hardware design of the module and the components of the module enhanced with specific additional details about the software and electronics in order to show the performances of the system.

6.1 Description of the Module

An assembly process module should include some modular task units for the realization of a predefined assembly process. These modules can be listed as follows;

- *Manipulators* – These manipulators can be mechanisms having several degrees of freedom, or simple linear or rotational stages according to the complexity of the task. They should have the necessary working space and high precision and accuracy for the assembly task. They should be equipped with necessary end effectors in order to manipulate the parts to be assembled. Types of the actuators to be used for the manipulators are subject to change considering the priority of the given criteria. The high precision necessity may limit the working range of the actuators and vice versa. In that context according to the necessities of the application, the actuator selection should be considered carefully. In some cases, when high precision and the working range necessities can’t be provided by a single actuator, a coarse fine positioning system can be configured using a combination of different actuators.

- *Vision Sensors* – A vision sensor is necessary for the position/orientation determination of the parts because of the fixturing limitations of the small parts. The parameters to be considered for the selection of a vision sensor are explained in details in Section 4.2.4. It can also be used to provide online visual feedback during the assembly operation for the generation of a fully automated assembly operation. According to the needs of the assembly process multiple systems can be used.
- *Carriage Units* – Since the module is designed to be a process module, assembly processes with different complexities should be implemented within the module. In that context the module may consist of several task units realizing the specific parts of the assembly. The parts or sub-assembled parts should be transferred in between the task units for the flow of the assembly process. A carriage unit is necessary in order to transfer parts in between the task unit stations. The structure of the carriage unit can be configured according to the layout within the module and the location of the stations. The positioning precision and the velocity of the carriage unit should be considered before the selection for the module.
- *Sensors* – In order to realize the assembly flow in a healthy manner, some sensors may be necessary for different purposes. These sensors may be related to the assembly process of the parts, positioning of the parts/trays on the carriage unit, etc.

In order to provide the bilevel modularity concept introduced for the microfactory modules, each task module necessary for the assembly operations should be designed in a modular way for easy integration of the unit inside the POM in order to realize different assembly processes.

The dimensions of the module are determined according to the dimensions of the electronics necessary for the system and the hardware modules that are used in the system. The dimensions allow easy integration of the modules inside which allows space for the integration of necessary number of task units inside the module. The module is built using aluminum sigma profiles for keeping the weight of the module low and the structure of profiles allow easy integration of any necessary hardware to any place in the module using necessary fixtures. The dimensions of the module are shown in Figure 6-1.

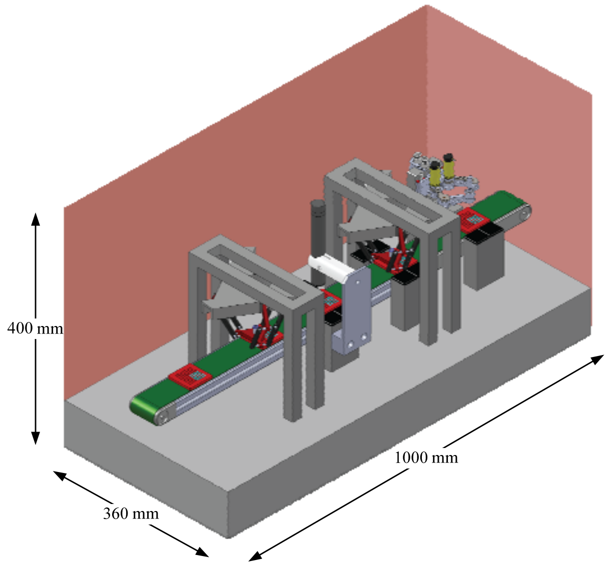


Figure 6-1 – Dimensions of the Assembly Module

6.2 Components of the Assembly Module

The necessary components for an assembly process are defined in the previous section. In that context, for each item listed, system necessities for the assembly module are determined and the modules are developed to satisfy these necessities. The submodules of the assembly module are explained in details in the following sections. In addition to the assembly module components developed, a microassembly and micromanipulation module developed as a part of a whole microassembly workstation is explained since there is the possibility of the integration of the manipulation module with the small modular vision system in the microfactory assembly module. The integrability of this module to the microfactory process oriented modules can be given as a proof for the reconfigurability of the module structure for different processes.

The explanation of the whole microassembly workstation with all features as a summary is given in the following section to show the functional features of the system. Then the integrability issues of the microassembly and micromanipulation module to the microfactory POMs is discussed. The development of the microassembly workstation involves several prototypes but the final prototype is given in the following section since it involves the early stages of this PhD study.

6.2.1 Microassembly and Micromanipulation Module

There are significant differences between the necessary requirements for assembly in the macro world and the microworld. The main difference is the positional accuracy required for the assembly machines. For the assembly in microworld submicron precision is necessary for the manipulators. Positioning at the microscale becomes considerably more difficult since accurate sensing of the true output is more difficult, and link flexibility can induce residual structural vibrations.

Another issue that has great significance in micromanipulation is the effect of force scaling. While the size of the objects become smaller, inertial forces scale down faster than adhesive forces as the inertial forces depend on the volume of the object and adhesive forces depend on the surface of the object. As a result of the effect of adhesive forces, the assembly operation becomes more difficult since releasing the object after gripping becomes a significant problem.

Differences between the assembly in macro and micro worlds forms the basic requirements for the design of a microassembly system. The main differences can be defined as accuracy, speed, repetitiveness and reliability. The main challenges in order to provide the requirements for the design of a miniaturized assembly system for the assembly of small parts can be listed as follows;

- Mechanical structure configuration of the microassembly workstation is a challenging issue since miniaturization requires development of a whole range of new miniature servo systems and measurement systems with high accuracy and repeatability.
- Robust control system design is of great interest in the microassembly. Modeling and control especially vision assisted control, become more critical in microassembly as the accuracy requirements increase and the size of parts decreases.
- Measurement of position and orientation of microparts is a complicated task that requires a sophisticated measuring system.
- A modular design of both the hardware and the software of the assembly system is an important design consideration for the scalability issues.

These issues are addressed in the design and realization of a microassembly workstation structure. The microassembly workstation is designed in such a reconfigurable and open architecture manner since it will be used as a research tool for the investigation of problems in the microworld. That design necessity makes the modularity of the system both in software and hardware system significantly necessary since the workstation should be configured according to the requirements of the application to be implemented using the system.

6.2.1.1 System Description

The development of such a workstation includes the design of (i) a manipulation system consisting of motion stages providing necessary travel range and precision for the realization of assembly tasks, (ii) a vision system to visualize the microworld and the determination of the position and orientation of micro components to be assembled, (iii) a robust control system and necessary mounts for the end effectors in such a way that according to the task to be realized, the manipulation tools can be easily changed and the system will be ready for the predefined task.

The overall functional structure of the workstation is depicted in Figure 6-2. The developed workstation is providing environment (positioning and vision systems) allowing wide range of the tasks to be performed by changing the endeffector tools attached to the end of each manipulator system. The overall mechanical motion has 9 DOF in manipulation and 3 DOF in vision system. According to the endeffectors to be used, the number of DOF is subject to change in the system.

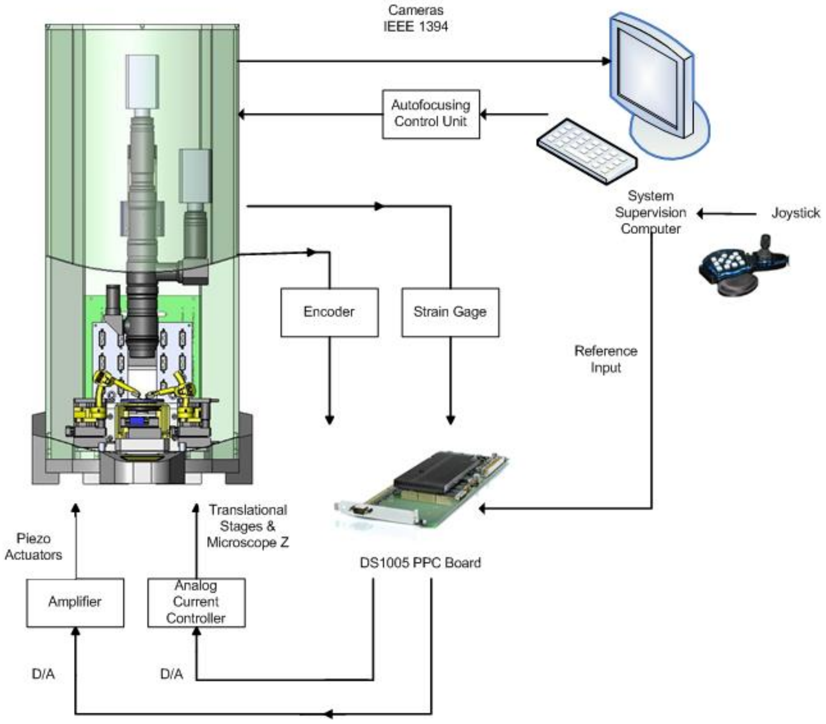

Cameras
IEEE 1394
Autofocusing Control Unit
System Supervision Computer
Joystick
Encoder
Strain Gage
Reference Input
DS1005 PPC Board
Piezo Actuators
Translational Stages & Microscope Z
Amplifier
Analog Current Controller
D/A
D/A

Figure 6-2 – System Configuration

6.2.1.1.1 Manipulation System

Manipulation system consists of two 3 DOF tool holder micromanipulator stages. Each manipulator consists of three linear stages configured as a Cartesian xyz system with 7 nanometers design resolution. The system presented in [13] which was the first prototype of the microassembly workstation has only one manipulator stage configured as a coarse and fine positioning stage. This configuration is changed in that workstation since the desired precision and travel range can be provided by using the linear stages. A second manipulator system with the same configuration is added to the system in order to perform more complex manipulation and assembly tasks with coordinated motion [12]. For example; a cell can be manipulated by a probe while it is being held or supported by means of a suitable end effector.

The system also has a 3 DOF sample precision positioning system (x,y,θ) which provides the usage of the substrate surface more effectively by moving the different regions of the substrate into the field of view of the microscope. Rotational stage is designed over a xy Cartesian positioning system with the resolution of 45 nano degrees. The design of the stage also allows backlighting with a gap opening of 20mm.

End effectors and necessary fixtures are used interchangeably in the system. Microgrippers, probes and other manipulation tools can be the matter of choice and necessary fixtures are designed to be easily integrated to the system. The whole system is placed onto an actively controlled damping table in order to get rid of environmental vibrations.

6.2.1.1.2 Vision System

Vision system is designed to provide the system and the operator clear visualization of the microworld, position and orientation of the parts to be manipulated. Integrated with the hardware system, user interface designed for the vision system utilizes coarse and fine views of the workspace, enables adjusting necessary magnification values for the task to be performed, autofocusing to provide clear images and determine the depth information. It also allows the realization of automated tasks with the utilization of visual feedback by determining the relative distances between the regions of interest and supplying the necessary information to the motion stages.

The optical microscope used in the workstation as the visual system is selected and configured according to the needs of a microassembly and manipulation system. It is configured in such a way that with two different optical paths, one with constant magnification to provide global view of the workspace and the other with extra magnification for the detailed view of the workspace. By enabling such a feature, positions of microparts scattered all over the sample plate can be determined and transferred to the assembly point of interest. On the other hand, by means of magnified view, assembly or manipulation of parts can be realized precisely with more accurate handling of the parts to be manipulated. Focus and magnification adjustment are controlled with stepper motors in order to fully automate the operations performed in the system. Illumination is the key issue for the vision system. As the assembly tasks require image processing algorithms for the detection of the parts to be manipulated and information about the geometries of these parts, the illumination techniques should be considered carefully. The system is equipped with two light sources, one providing backlight illumination by means of a RGB Led Illuminator and an upper illuminator from the microscope's vision path. Both illumination systems can be controlled from the system computer.

6.2.1.2 Experiments and System Performance

For testing of the reliability of the system several experiments are implemented in different modes of operation; tele-operated, semi-automated and fully automated by means of visual based schemes. Experiments are realized using polystyrene microspheres with diameters of approximately 50 μm and using various manipulation tools. Tele-operated microassembly is realized in two different ways; by giving commands on the screen with mouse clicks or by means of a joystick. Semi-automated micro assembly involves the intervention of the operator to some extent. The operator only chooses the particle to be manipulated and the destination point where the particle is to be moved, the rest is executed automatically.

By using the GUI developed for the workstation, in the tele-operation mode, using a sharp tungsten manipulator to push the micro spheres with a diameter of 50 μm, initial letters of Sabanci University is formed and shown in Figure 6-3.

Several conventional visual servoing schemes are also implemented for various microsystem applications. An image based visual servoing algorithm using optimal control penalizing the pixel error and the control signal magnitude is implemented to form a line pattern by pushing 50 μm diameter polystyrene spheres. The line pattern formed using the probe is shown Figure 6-4. In order to locate the microspheres precisely, a feature extraction algorithm to detect the tip of the probe and the spheres is developed. Moreover the trajectory planning is designed to avoid the obstacles and determine the priorities of the particles to be located.

The workstation is designed in such a way that various types of end effectors can easily be adapted to the system and used as the handling tools. Different types of microgrippers are used in order to implement pick place experiments in the system as shown in Figure 6-5. The workstation is also used for cell manipulation experiments using micropipettes, probes, etc.

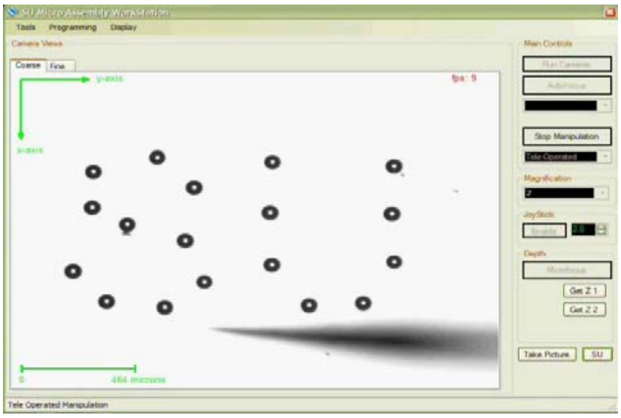


Figure 6-3 - SU Pattern Formation.

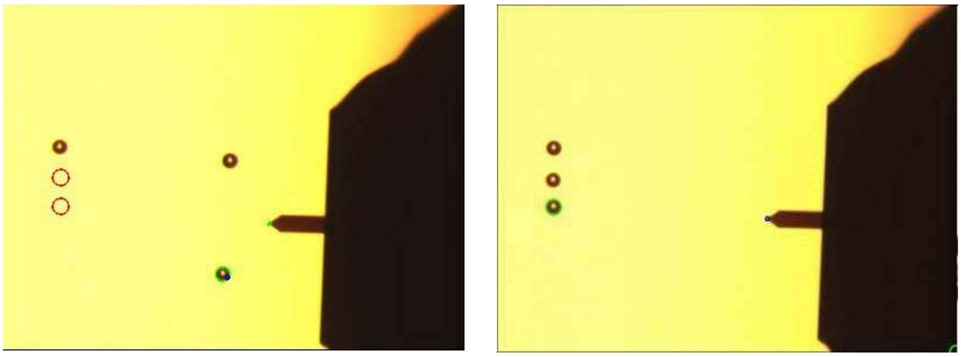


Figure 6-4 - Line Pattern Formation Using Visual Based Schemes.[12]

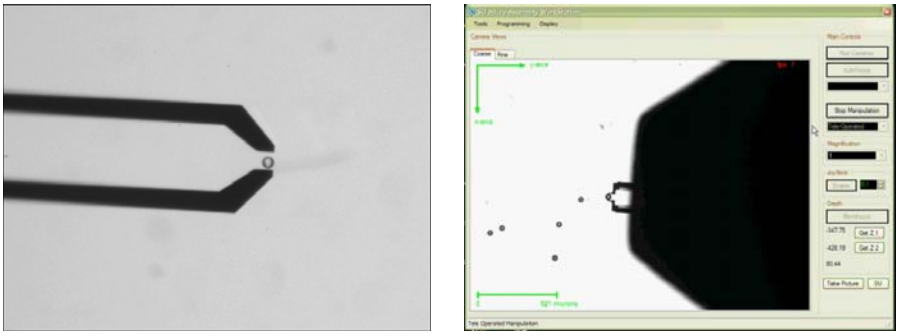


Figure 6-5 - Microgripper Experiments.

The manipulation unit of the microassembly workstation is designed in a modular way in order to allow easy removal of the system as a unit and integration to other systems. The manipulation system itself consists of 2 manipulators and a sample platform each having 3 DOFs. The control electronics of the microfactory module is capable of handling 12 DOFs with the FPGA system and can be extended with the usage of another control electronics that is capable of handling more DOFs. The tray of the microassembly and micromanipulation module is designed to be easily removable. With the integration of a tray extraction manipulator or system, the module can be easily integrated to the microfactory modules. The optical microscope used in the microassembly workstation is a large one, however the vision sensor used in the assembly module can be integrated as a vision sensor to the unit since it allows integration of extra lenses for increasing the magnification which will be satisfactory enough for the visualization of the microparticles.

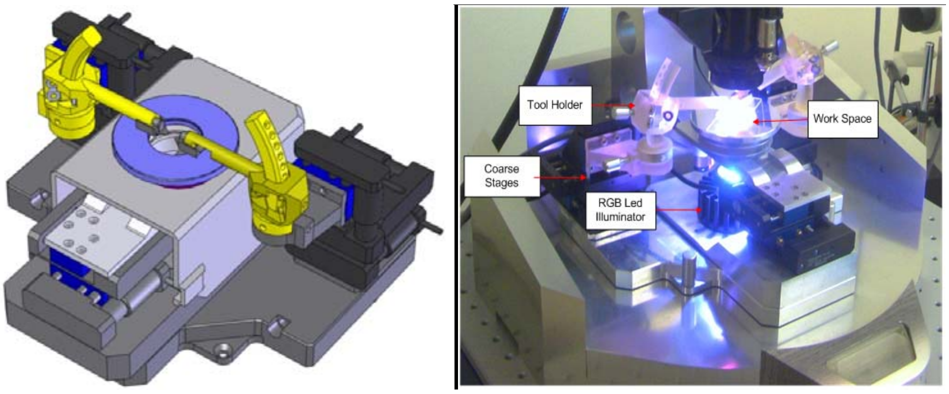


Figure 6-6 – Microassembly and Micromanipulation Module

The dimensions of the microassembly and micromanipulation module are given in Figure 6-7. The size of the module and the structure of the POMs allow it to be placed in the microfactory module. With the integration of the module the POM can be configured as a microassembly and micromanipulation POM.

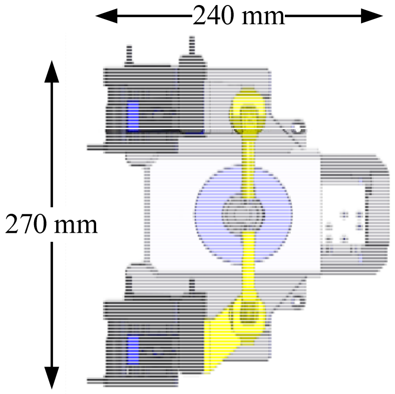


Figure 6-7 – Dimensions of the Microassembly and Micromanipulation Module

6.2.2 Delta Robot

The Delta robot (a parallel kinematics robot) was invented in the early 1980's by Reymond Clavel at the École Polytechnique Fédérale de Lausanne (EPFL, Switzerland). The purpose of this new type of robot was to manipulate light and small objects at a very high speed which was a crucial industrial need at that time. In 1987, the company Demaurex purchased a license for the Delta robot and started the production of Delta robots for the packaging industry. In 1991, Reymond Clavel presented his doctoral thesis "Conception d'un robot parallèle rapide à 4 degrés de liberté" and received the golden robot award in 1999 for his work and development of the Delta robot. Also in 1999, ABB Flexible Automation starts selling its Delta robot, the FlexPicker. By the end of 1999 the Delta robots are also sold by Sigpack Systems. Soon enough, several other companies like Bosch and Festo produced their own Delta robots.

This parallel mechanism has several crucial advantages over other solutions like low inertia, accuracy, speed and stiffness. On the other hand, Delta robot has limited workspace for which their mechanical designs should be optimized to have homogeneous precision distributions. Also, Delta robots are designed to manipulate proportionally small objects with high speed and accuracy since they have small payloads. The use of three parallelograms in the design of Delta robot restrains completely the orientation of the mobile platform which remains only with three purely translational degrees of freedom. It is generally used as a pick place robot mostly in the food industry with its fast and precise moving capabilities. Delta Robot that is implemented within the context of microfactory modules is a miniaturized version of the existing ones already working in the industry. The workspace of the miniaturized version is limited to 40mm cube and the kinematic parameters of the robot are determined accordingly.

6.2.2.1 Design Issues

Delta robot, shown in Figure 6-8, consists of a traveling plate which is connected to the base by three identical parallel kinematic chains and each of them is actuated by a revolute motor mounted on the fixed base plate. Each chain consists of an upper arm, actuated by the revolute motors and a lower arm each of which has the formation of a parallelogram formed by links and spherical joints. The motion is transmitted to the traveling plate from the actuated upper arms through the lower arms. The parallelogram structure of the lower arms assures the parallelism of the traveling plate to the fixed base plate.

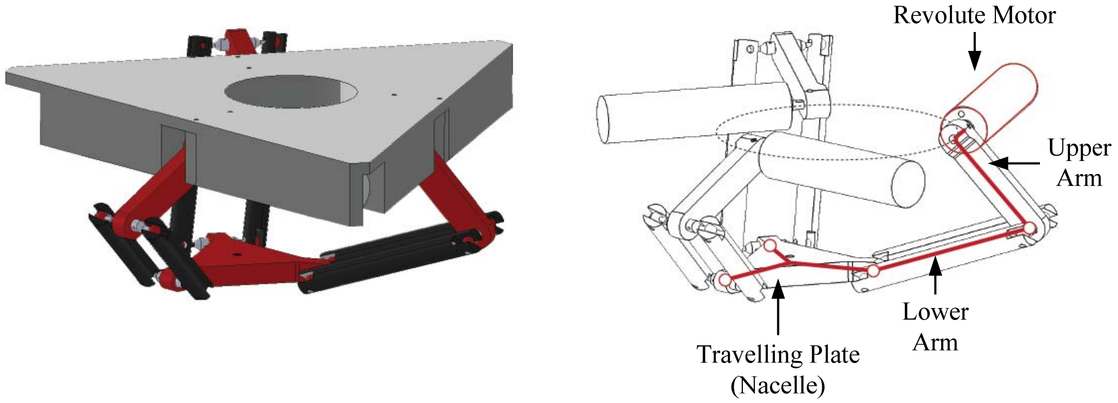


Figure 6-8 – Delta Robot

For the design of a Delta robot, initially the kinematic parameters of the robot like the link lengths should be determined. The design parameters are determined for a desired workspace of 40 mm cube with the optimization method explained in the following section.

6.2.2.1.1 Optimization

For the design of the miniature Delta robot aimed to be a realized as a part of the microfactory assembly module for a predefined workspace of 40mm cube, optimization technique described in [41] is implemented for the derivation of the design variables.

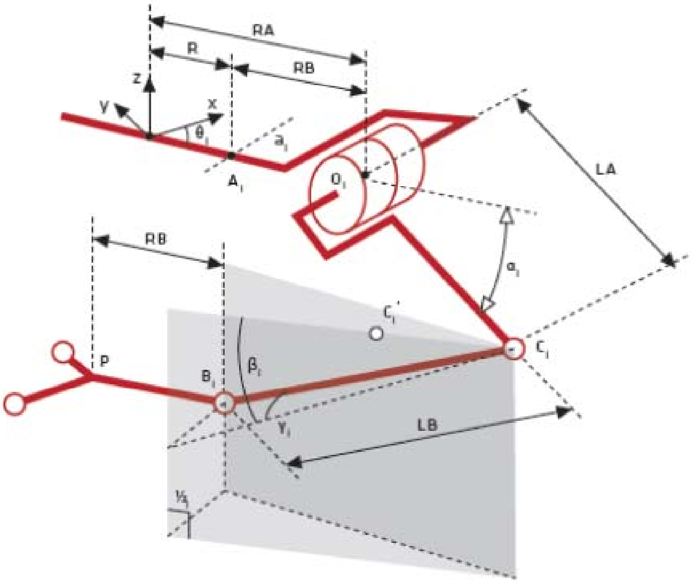


Figure 6-9 – Reymond Clavel’s Kinematic Chain Model

For a given workspace, the link lengths and the size of the base with respect to the size of the end effector are the three important design variables for the robot. The design variables are shown as depicted in [41]. The term “base” is representing the triangle formed by the three arms of the robot which is the fixed part of the robot unlike the end-effector. These parameters actually determine the workspace of the Delta robot.

Besides dimensional variables of the robot, there should be one more parameter that relates the both workspaces, because determining the optimum workspace of the robot is not enough as long as those both workspaces are not considered together properly. The parameter that relates both workspaces is simply the distance between the base and the pre-determined workspace. Eventually there are 4 variables that should be determined by the optimization process. These parameters are;

L_A (Upper Arm Length), L_B (Lower Arm Length), R (R_A-R_B) and H (the distance between the base and the pre-determined workspace)

The most challenging part of the optimization process is to determine the objective function. It is a function such that the values of the variables which make the function value minimum are the ones which give the highest sensitivity in motion.

First of all, the mathematical model of the workspace of the robot is the following;

$$h_j(X_p, Y_p, Z_p) = ((X_p cos\theta_j + Y_p sin\theta_j - R)^2 + (X_p sin\theta_j - Y_p cos\theta_j)^2 + Z_p^2 + L_B^2 - L_A^2)^2 - 4L_B^2((X_p cos\theta_j + Y_p sin\theta_j - R)^2 + Z_p^2) \leq 0 \quad (6\text{-}1)$$

The equation above represents a volume in a space. The boundary of this volume is defined by $h_j(P) = h_j(X_p, Y_p, Z_p) = 0, (j = 1,2,3)$, where P is a point with the coordinates X_p, Y_p, Z_p. Note that, when h_j applied to a point, it can be used as a measure of some kind of distance of this point with respect to the surface defined by $h_j = 0$. Moreover, the function h_j changes its sign depending on which side of the surface the point is located. Therefore minimizing the function $|h_j(I, P^k)|$ with respect to $I = [L_A, L_B, R, H]$, is equivalent to finding a surface (which depends on I) closest to the point P^k. In our case, we are looking for a volume bounded by three surfaces; therefore one has to minimize the function $F(I, P^k) = |h_1(I, P^k) + h_2(I, P^k) + h_3(I, P^k)|$.

Eventually, we should determine such points that construct a desired workspace, and every chosen point should be plugged into the function above and minimized. So the objective function is constructed as

$$F(I, P^k) = \sum_{k=1}^{Npt} \sum_{j=1}^{3} |hj(I, P^k)| \quad (6\text{-}2)$$

The constraints of this optimization problem are such that, all the points P^k that will be chosen to determine the workspace must be contained by the workspace of the Delta robot. In other words, every point of the pre-determined workspace must be reachable by the robot.

When we look at the mathematical description of the workspace of the robot, we can conclude the following situations for an arbitrary point P which has the coordinates of X_p, Y_p, Z_p;

- P is inside the workspace if $h_j(X_p, Y_p, Z_p) < 0 \; for \; j = 1,2,3$
- P is at the boundary of the workspace if $h_j(X_p, Y_p, Z_p) \leq 0 \; for \; j = 1,2,3$ and $h_j(X_p, Y_p, Z_p) = 0$ for $j = 1 \; or \; j = 2 \; or \; j = 3$

- P is outside the workspace iff $h_j(X_p, Y_p, Z_p) > 0\ for\ j = 1,2,3$

According to the conditions above it can be noted that all the points in the pre-determined workspace must sustain the first case. By using penalty function method, the constraints can be plugged into the objective function as follows;

$$F_{penalty} = \sum_{k=1}^{Npt} \sum_{j=1}^{3} \wp j(I, P^k) \tag{6-3}$$

$$\wp j(I, P^k) = \begin{cases} 0\ if\ hj(I, P^k) \leq 0 \\ cf\ if\ hj(I, P^k) > 0 \end{cases} \tag{6-4}$$

As a result, the objective function is upgraded as;

$$F_{objective} = F + F_{penalty} \tag{6-5}$$

Using heuristic based algorithm methods are more suitable for that problem since the mathematical model of the problem is complex and we also need to deal with the local minimum points. Genetic Algorithm as the most common and reliable heuristic algorithm is executed for the optimization using the Genetic Algorithm Solver and Direct Search Toolbox in MATLAB. The objective function is defined as described above and initial parameters for the algorithm with the boundaries of the target variables are given as input;

Number of Variables: 4

Bounds: Lower = 40, 50, 40, 20; Upper = 80,100,100,30

Population Type: Double Vector

Population size: 100

Fitness Scaling: Rank

Crossover Function: Scattered

Migration Direction: Forward

Stopping Criteria: Generations = 150

The parameters (L_A, L_B, R_A , R_B) are the four variables to be obtained and after an iterative process the results obtained are:

L_A = 40 mm L_B = 67.863 mm R_A = 40 mm R_B = 29.997 mm

Using the design parameters obtained, the workspace coverage analysis is realized using the kinematics of the robot which is given in the following sections. As it can be seen in Figure 6-10, the design can reach to most of the workspace and covers it as much as possible without wasting its range outside of the workspace.

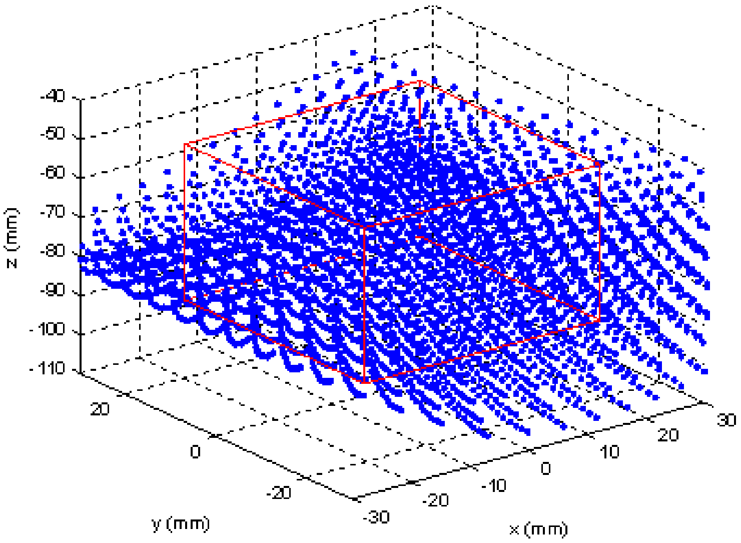


Figure 6-10 – Workspace Coverage

6.2.2.1.2 Kinematics of Delta Robot

Parallel robots provide the advantages of high stiffness, low inertia and high speed capability at the expense of smaller workspace, complex mechanical design, difficult forward kinematics and control algorithm. The kinematics of the parallel mechanisms is more complex when compared to the serial ones and needs more computational power. Different models have been proposed for the geometry model of the Delta Robot [41]-[45]. The model developed by R. Clavel in [32] is based on the principle that the distance between B_i and C_i (end of the arm) must remain constant and equal to the length of the parallelogram structure. This model is used for the computation of the forward and inverse kinematics of the Delta robot. Descriptions of the significant variables used in the kinematic calculations as shown in Figure 6-9 are;

$\boldsymbol{R_A}$: The distance between the center of the base and the axis of rotation of the actuator.

$\boldsymbol{R_B}$: The distance between the center of the “nacelle” (the travelling plate) and the rotational axis of the bottom spherical joints.

$\boldsymbol{L_A}$: The length of the upper link (active joint)

$\boldsymbol{L_B}$: The length of the bottom link (passive joint)

$\boldsymbol{\alpha_i}$: The angle between the upper arm and the plane of the base, considering that the base is horizontal, by convention α_i is negative when the arm is located below that horizontal plane.

$\boldsymbol{\beta_i}$: The angle between the plane of parallelogram and the plane of the base

$\boldsymbol{\gamma_i}$: The angle between the vertical plane shown in Fig. 1 and one of the bars of the parallelogram

$\boldsymbol{\theta_i}$: The angular position of the corresponding links depending on the selected base x-axis..

The determination of the coordinates x, y, z of point P from the joint coordinates α_i (forward kinematics) and calculations for the inverse operation are based on the following reasoning: Each point O_i is moved in A_i and each point B_i in P by three translations with the amplitude of R_B; the distance OA_i: $R_A - R_B = R$. The point P is thus the center of a sphere of radius L_B; points A_i are centers of circles of radius L_A on planes π_i and points C_i (translated) are given by the intersections of three circles of radius L_A with the sphere of radius L_B centered at P.

The coordinates of the points C_i:

$$[(R + L_A cos\alpha_i)cos\theta_i, (R + L_A cos\alpha_i)sin\theta_i,\ L_A sin\alpha_i] \tag{6-6}$$

The equation of the sphere with the center $P(x, y, z)$ with radius of L_B:

$$(X - x)^2 + (Y - y)^2 + (Z - z)^2 = L_B^2 \tag{6-7}$$

Putting the C_i coordinates into the sphere equation;

$$[(R + L_A cos\alpha_i)cos\theta_i - x]^2 + [(R + L_A cos\alpha_i)sin\theta_i - y]^2 + \ [L_A sin\alpha_i - z]^2 \tag{6-8}$$

which is

$$\begin{aligned} x^2 - 2x(R + L_A cos\alpha_i)cos\theta_i + y^2 - 2y(R + L_A cos\alpha_i)sin\theta_i + z^2 \\ - 2zL_A sin\alpha_i = L_B^2 - L_A^2 - R^2 - 2RL_A cos\alpha_i \ for\ i = 1,2,3 \end{aligned} \tag{6-9}$$

These three equations can be resolved for the coordinates x, y, z for the forward and inverse kinematics.

Forward Kinematics equations for the Delta robot are depicted below;

$$\begin{gathered} D_i = -L_B^2 + L_A^2 + R^2 + 2RL_A cos\alpha_i \\ E_i = 2(R + L_A cos\alpha_i)cos\theta_i \end{gathered} \tag{6-10}$$

$$F_i = 2(R + L_A cos\alpha_i)sin\theta_i = E_i tan\theta_i$$

$$G_i = 2L_A sin\alpha_i$$

$$H_1 = E_1G_2 - E_1G_3 - E_2G_1 + E_2G_3 + E_3G_1 - E_3G_2$$

$$H_2 = -E_1F_2 + E_1F_3 + E_2F_1 - E_2F_3 - E_3F_1 + E_3F_2$$

$$H_3 = -E_1D_2 + E_1D_3 + E_2D_1 - E_2D_3 - E_3D_1 + E_3D_2$$

$$H_4 = F_1D_2 - F_1D_3 - F_2D_1 + F_2D_3 + F_3D_1 - F_3D_2$$

$$H_5 = -F_1G_2 + F_1G_3 + F_2G_1 - F_2G_3 - F_3G_1 + F_3G_2$$

Then

$$\begin{gathered} x = z\frac{H_5}{H_2} + \frac{H_4}{H_2} \\ y = z\frac{H_1}{H_2} + \frac{H_3}{H_2} \end{gathered} \tag{6-11}$$

Putting (2) and (3) in (1) for i=1

$$z = \frac{-M \pm \sqrt{M^2 - 4LQ}}{2L} \tag{6-12}$$

where

$$\begin{gathered} L = \frac{H_5^2 + H_1^2}{H_2^2} + 1 \\ M = 2\frac{H_5H_4 + H_1H_3}{H_2^2} - \frac{H_5E_1 + H_1F_1}{H_2} - G_1 \\ Q = \frac{H_4^2 + H_3^2}{H_2^2} - \frac{H_4E_1 + H_3F_1}{H_2} + D_1 \end{gathered} \tag{6-13}$$

Inverse Kinematics equations are depicted below;

$$Q_i = 2xcos\theta_i + 2ysin\theta_i \tag{6-14}$$

$$S = \frac{1}{L_A}(-x^2 - y^2 - z^2 + L_B^2 - L_A^2 - R^2)2xcos\theta_i + 2ysin\theta_i \tag{6-15}$$

$$tan\frac{\alpha_i}{2} = \frac{2z \pm \sqrt{4z^2 - 4R^2 - S^2 + Q_i^2\left(1 - \frac{R^2}{L_A^2}\right) + Q_i(-2\frac{RS}{L_A} - 4R)}}{-2R - S - Q_i(\frac{R}{L_A} - 1)} \tag{6-16}$$

The analytical method offered by Clavel [32] is used for the simulation purposes in Matlab. Both forward and inverse kinematic solvers yield consistent results.

6.2.2.2 Prototypes of Delta Robot

The kinematic design parameters of the Delta robot are obtained with the optimization process according to the predetermined workspace dimensions. Link lengths and other design parameters are determined and the mechanical design phase is completed with the resulting parameters. Before the implementation of the mechanical design, tests are performed to check if the robot covers the desired workspace without any problems in the simulation environment.

Three prototypes are designed and developed for the miniaturized Delta robot according to the performance test results. The problems are determined related to the design of the robot or the manufacturing of the parts and necessary enhancements are realized in order to increase the performance of the robot.

The CAD drawing and the pictures of the first prototype of the robot are shown in Figure 6-11 and Figure 6-12 respectively. After the design and realization of the Delta robot, performance of the robot is tested with experiments and during testing design problems are observed which mainly formed input for the design of a second prototype of the robot.

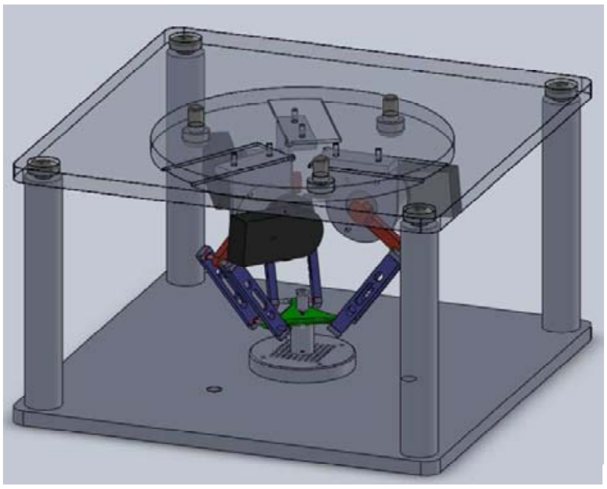


Figure 6-11 – First Prototype CAD

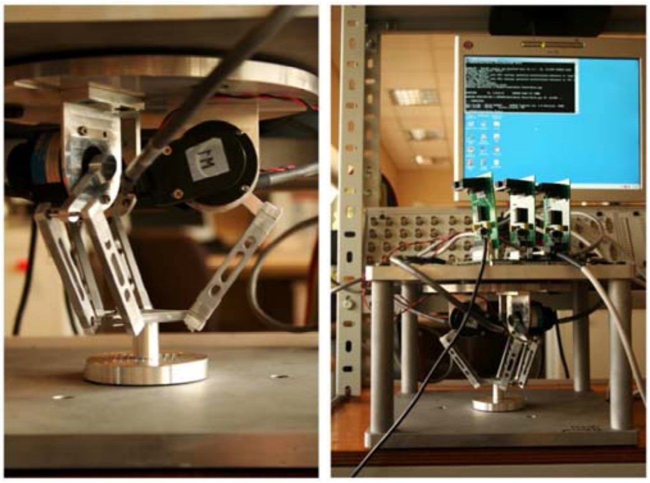


Figure 6-12 – Miniature Delta Robot (First Prototype)

Actuators which are selected to be used for the first prototype of the Delta robot are limited angle torque motors working in the range [-90 90] degrees and can give continuous torque in the range [-60 60]. The main reason for using direct drive motors is that direct drive motors can achieve high speeds and torque which is the signature of the Delta robot designs. For the testing of the Delta robot performance, dSPACE 1103 platform is used as the control platform. The drivers and the connection board are designed specifically for the dSPACE. The direct drive motors have no specific drivers available so that dedicated drivers are designed.

The major design criterion for the electronics is that the motors can take up to 3-3.5 A when they are close to their limits. Several designs are implemented, which includes building an H-bridge (which is the core of the power transfer to the motors) with transistors, IR2110 (high and low side driver) ICs, and finally using both channels of the L298 (dual full-bridge driver) ICs in parallel to withstand up to 4 A. The designed driver and the connection board with three drivers integrated are shown in Figure 6-13.

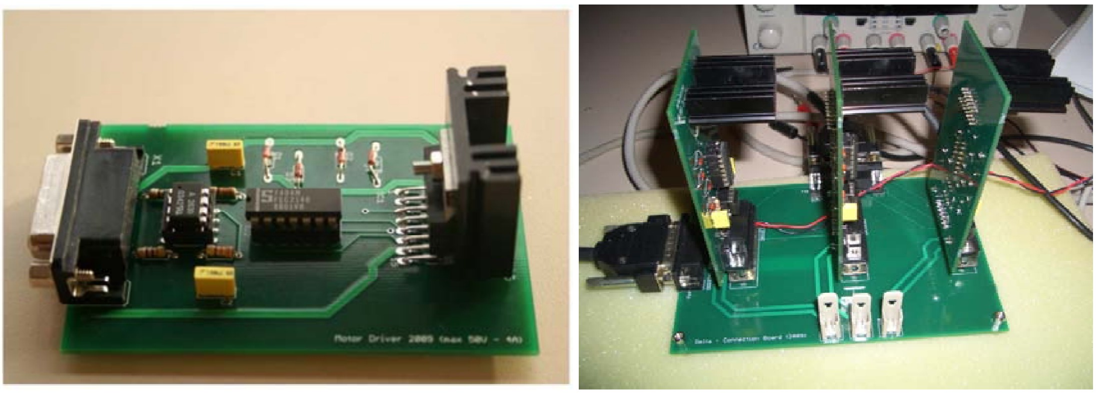


Figure 6-13 – a) Driver for the Direct Drive Motors b) Connection Board

In order to see the performance of the Delta robot in a pick and place scenario, a trajectory routine which tracks the corners of the workspace is implemented. This routine also contained the longest linear path in the workspace which is the diagonal between two opposite corners of the cube. This path is 69.282mm long for the 40mm cube case, and the robot is desired to perform at least 2 picks/sec. This means the end effector should be able to pass this diagonal at least 4 times in a second (move down to pick, pick up and move up, move down to the destination then move up for the next pick). The desired velocity to achieve such a task is estimated to be around 280 mm/s.

The following results contain the reference trajectory and the output coordinates of the end effector calculated with forward kinematics model from the encoder data. It is actually an open loop estimation for the coordinates which means the mechanical impurities and tolerances are ignored. For further correction over the output coordinates the position of the end effector needs to be measured using a suitable sensor and a calibration procedure is necessary for mechanical impurities and tolerances.

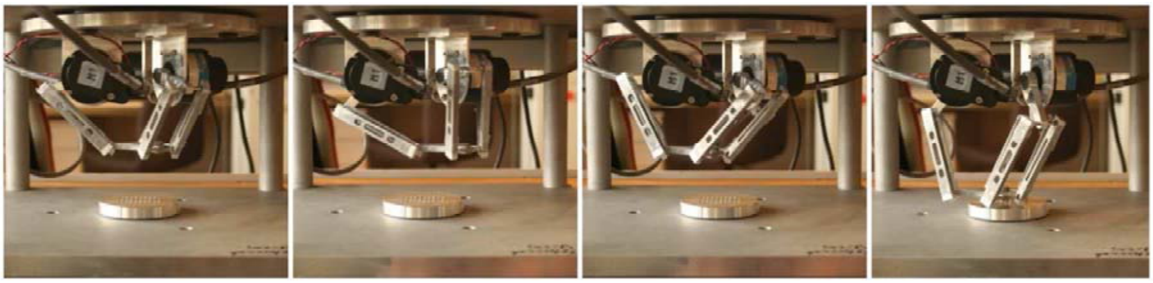


Figure 6-14 – The Routine Test

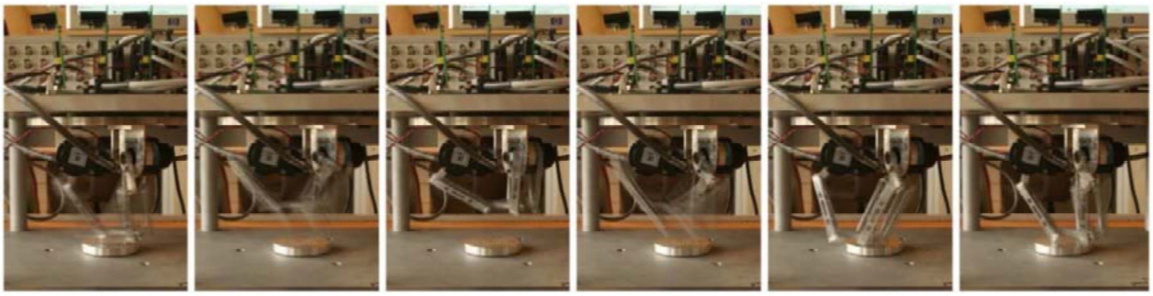


Figure 6-15 – Routine Test at Higher Speed

The experiments are realized at different speeds and the results are shown for the given trajectory routine at each speed. The error is calculated as the distance between the reference point and the calculated output as;

$$error = \sqrt{(x_r - x_0)^2 + (y_r - y_0)^2 + (z_r - z_0)^2} \tag{6-17}$$

First experimental trajectory routine tests are performed with 140 mm/sec and the results are shown in the following figures.

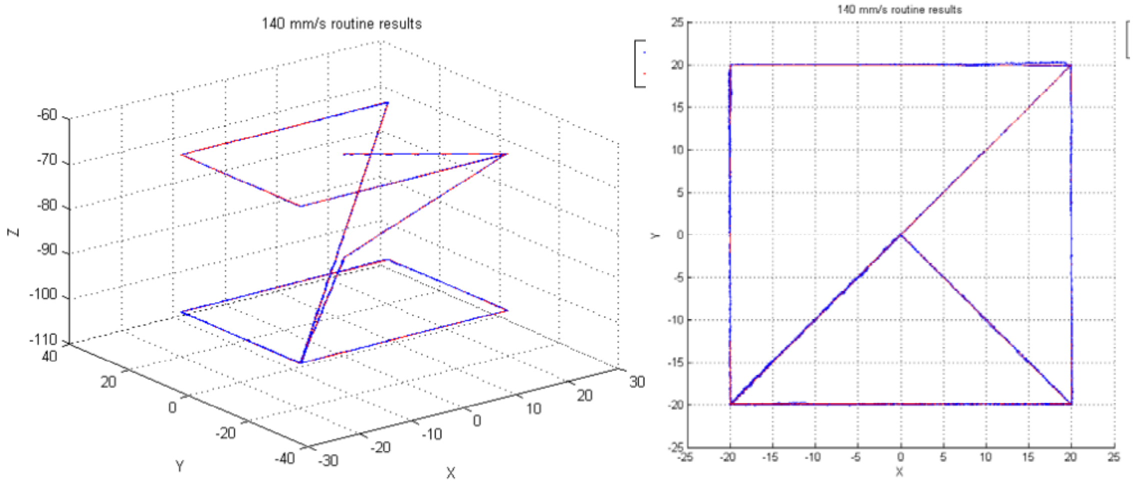


Figure 6-16 – Experimental results with 140 mm/s velocity reference. a) 3D View b) Top View

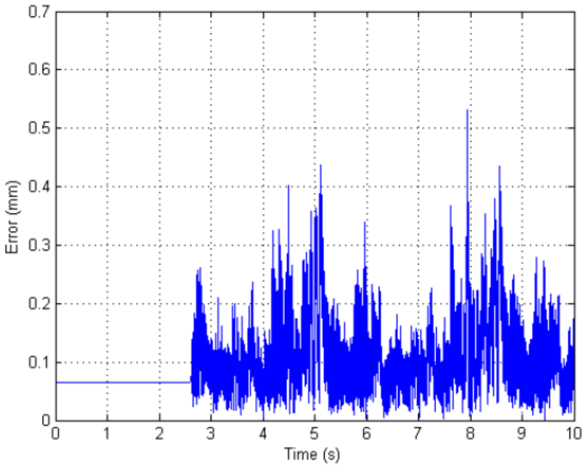


Figure 6-17 – Error against time (140 mm/s)

The motion starts at time 2.62 sec. so that the prior data is left out for the calculation of the min, max, mean and the standard deviation of the error data. The error values are calculated as;

min(error) = 2.9665e-006 mm

max(error) = 0.5304 mm

mean(error) = 0.1124 mm

std(error) = 0.0601 mm

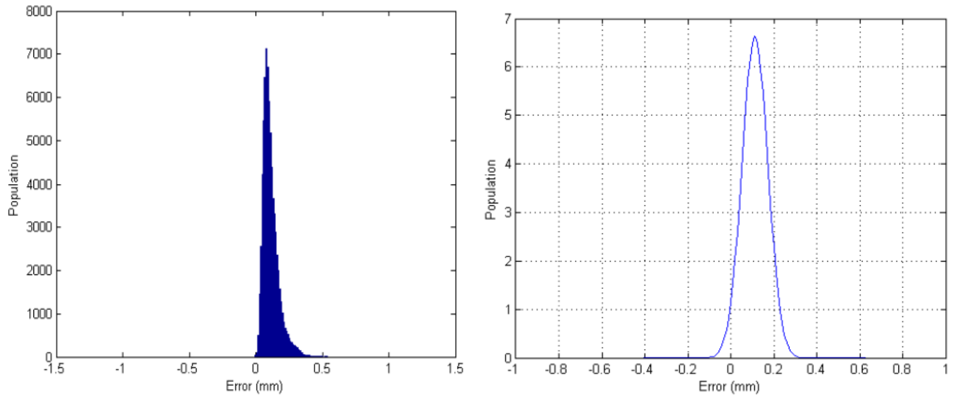


Figure 6-18 – a) Error Histogram (140 mm/s) b) Error PDF model (140 mm/s)

The same routine is executed with 280 mm/s end effector velocity and the results are given in the following figures.

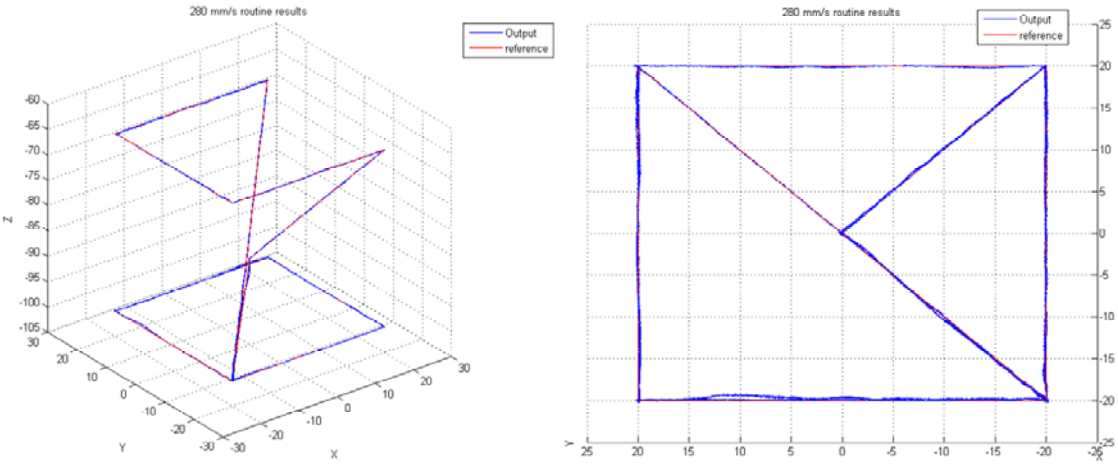


Figure 6-19 - The experimental results with 280 mm/s velocity reference. a) 3D View b) Top View

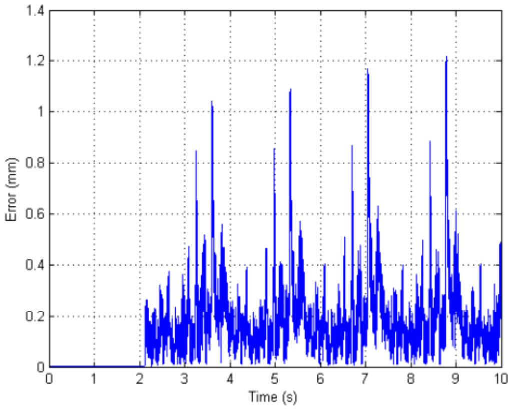


Figure 6-20 – Error against time (280mm/s)

The motion starts at time 2.13 seconds so that the prior data is left out for the calculation of the min, max, mean and the standard deviation of the error data.

min(error) = 0.0038 mm

max(error) = 1.2181 mm

mean(error) = 0.1999 mm

std(error) = 0.1504 mm

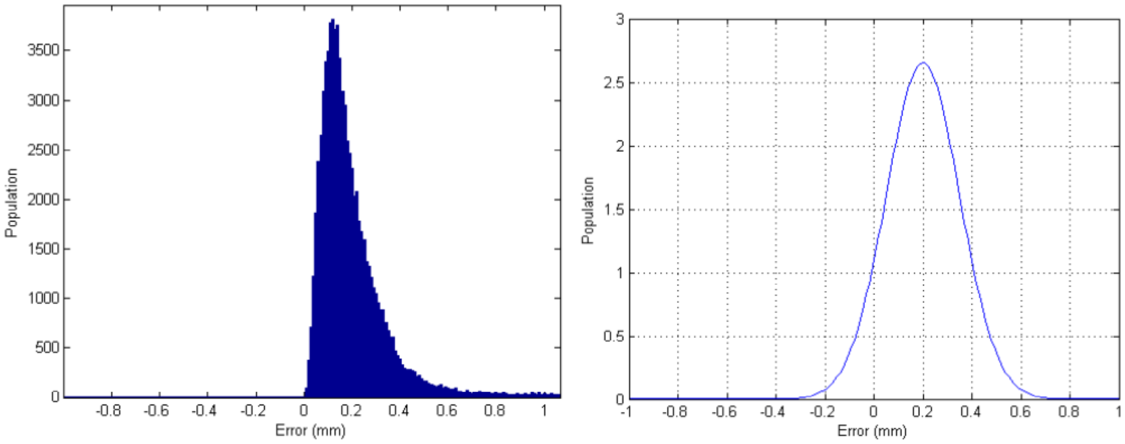


Figure 6-21 – a) Error Histogram (280mm/s) b) Error PDF model (280 mm/s)

It can be seen from the trajectory figures that the motion around the corners have a larger radius. The error against time graph shows a little cumulative behavior (peaks grow larger), this can be easily compensated by adding a sleep time at the corners, which is actually necessary for picking up the manipulated object.

The experimental trajectory routine figures with 350 mm/sec are given below.

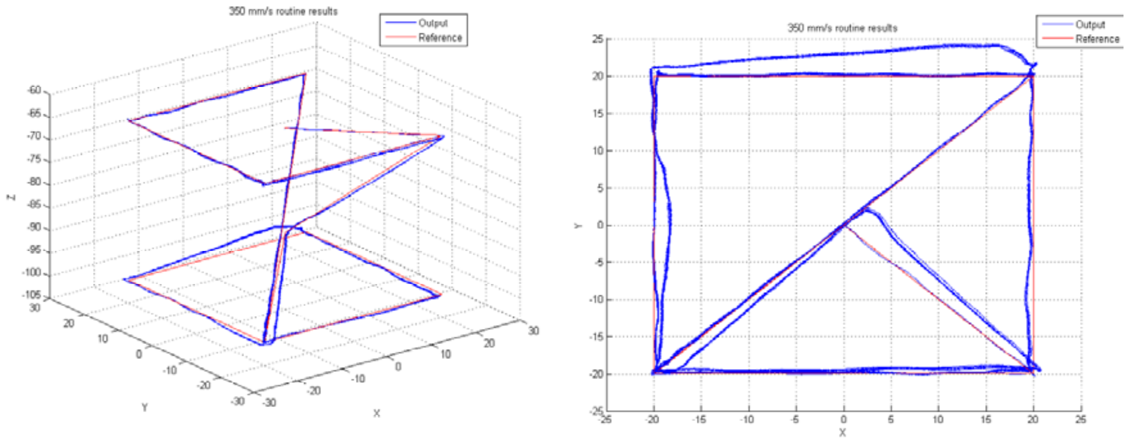


Figure 6-22 - The experimental results with 350 mm/s velocity reference. a) 3D View b) Top View

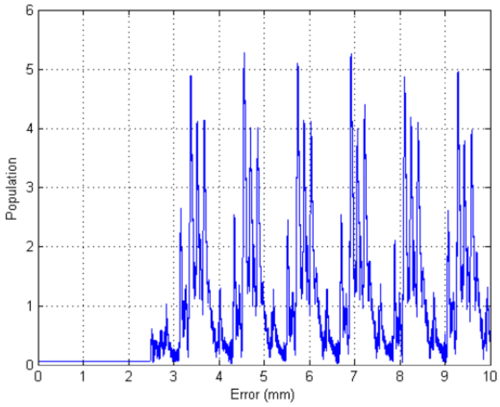


Figure 6-23 – Error against time (350mm/sec)

The motion starts at time 2.50 seconds so that the prior data is left out for the calculation of the min, max, mean and the standard deviation of the error data.

min(error) = 0.0219 mm

max(error) = 5.2661 mm

mean(error) = 1.2390 mm

std(error) = 1.1172 mm

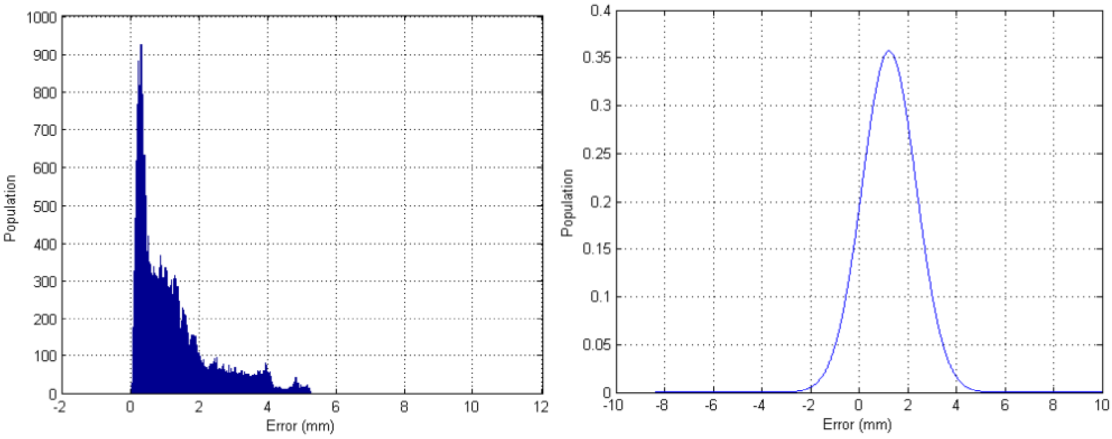


Figure 6-24 – a) Error Histogram (350mm/sec) b) Error PDF model (350mm/sec)

The error increases with the increasing velocity since the mechanical impurities affect the performance of the robot at high speeds. The experimental results show that a minimum of 2 pick and places per second can be performed with a mean error of 0.19mm which shows that major revisions are necessary. The longest linear path can be followed in 0.25 s with 280 mm/s speed. The encoder resolution is 0.09°/pulse which maps to different resolutions in the workspace, but it is estimated to be smaller than 0.015mm. The design and the parts manufactured have some major problems which affects the working performance of the Delta robot. As a result of the manufacturing deficiencies, the joints could not be mounted properly which creates an important problem for the motion. Resolution of the encoders suitable for these motors is not precise enough and since the motors are direct drive motors the resolution is limited just to the encoder pulses.

Since each arm of the parallelogram can move freely as a result of the joint design it affects the performance of the robot while working at high speed, the design must be revised considering these problems so that a second prototype is developed.

The second prototype of the Delta robot is shown in Figure 6-26. As it can be seen the lower arms of the robot are changed with the new ones which are designed by considering the manufacturing ease so that the errors resulting from that could be reduced. The new design of the joints is shown in Figure 6-25. For the joint design tolerances for the gaps where the ball joints fit in is given properly and designed for two different types of joints with different sizes of end spheres, necks and neck fit tolerances. Manufacturing facilities are considered for the design of the arms having the openings for the joints in order not to have manufacturing oriented problems in the new design.

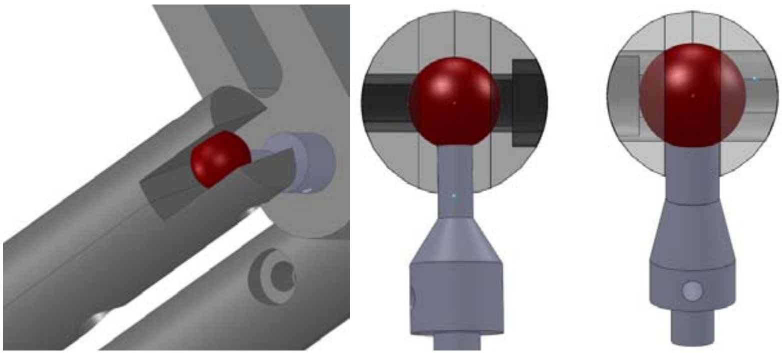


Figure 6-25 – Joint Design

The second prototype is developed with the new joint designs and the performance of the robot is tested with experiments. The endeffector position references are given to the robot as xyz coordinates and then the coordinates are transformed into joint space angles using the inverse kinematics equations of the robot, then the angular references are given to the motors where control is applied. The motor positions are given as input to the forward kinematics equations to generate the actual position of the robot. The following graphs are showing the references and the positions of the robot. XY plane motion showing the circular reference and the actual position of the robot is shown Figure 6-27 and each reference for the x and y vs. time graphs are shown separately in Figure 6-28. The error calculated as a sum of squared differences is given in Figure 6-29.

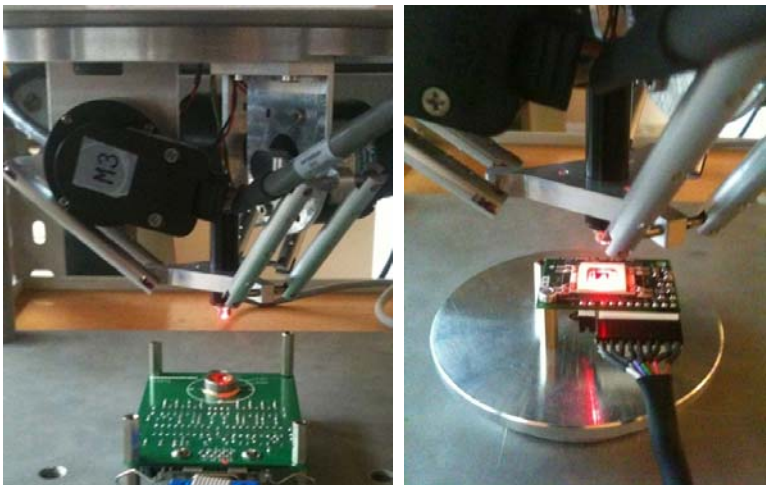


Figure 6-26 – (a) Second Prototype of Delta Robot (b) Testing the performance with external sensor

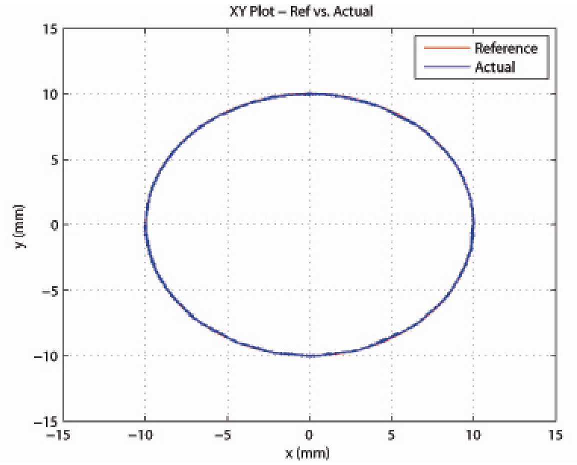


Figure 6-27 – XY Circular Graph (Second Prototype)

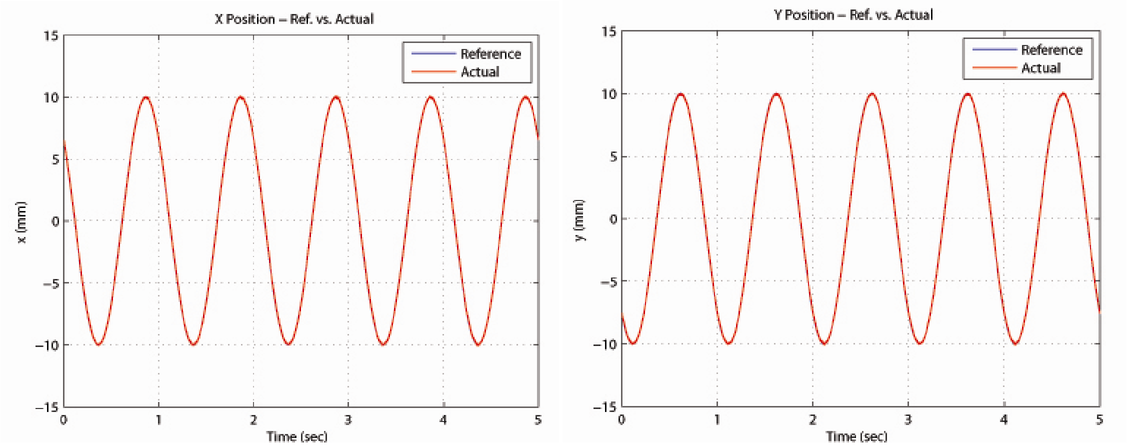


Figure 6-28 – (a) X Sinusoidal Reference (b) Y Sinusoidal Reference

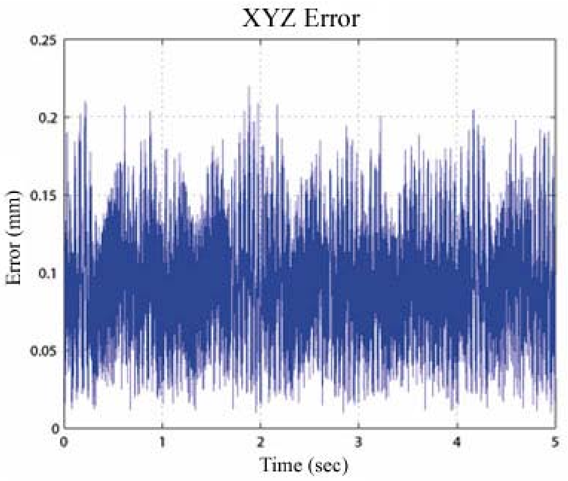


Figure 6-29 – Calculated Error

When compared with the first prototype, the performance in terms of precision of the robot is better than the first prototype. The maximum tracking error in the first prototype was 0.53 mm but now with the replacement of the lower arms in the new design it is lowered to 0.22 mm. These values are taken at high speeds of the Delta robot. However, the precision of the encoders integrated to the direct drive motors did not satisfy our needs so that another prototype is inevitable in order to improve the performance of the robot.

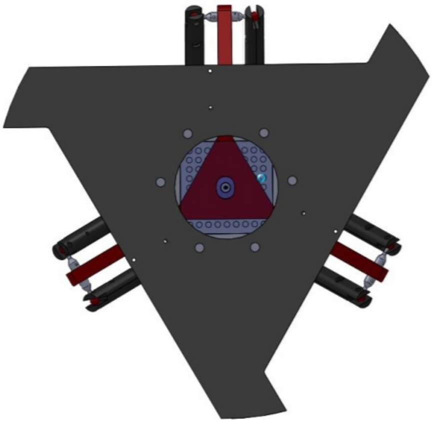


Figure 6-30 – Top View

The design of the new prototype is realized with the same dimensions for the kinematic parameters optimized for the previous design according to the prescribed workspace of 40mm cube. When compared with the previous design, the new design is more compact and some enhancements in the design are realized. The upper plate is designed in such a way that it allows the proper cabling for the motors and the end effector so that they will not prevent the motion of the manipulator. The opening on the upper plate allows the integration of any vision sensor for the position determination of the objects and the end effector.

As it can be seen in the bottom view the motors are embedded into the upper plate for the purpose of achieving a compact design.

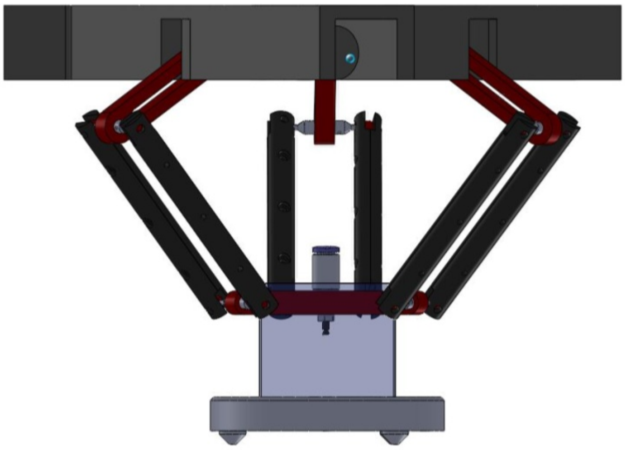


Figure 6-31 – Side View

Another issue that is considered for the new design is the compactness. In order to realize a more compact design, since it is not possible to find smaller direct drive motors, alternatives are considered and high speed brushless dc servomotors are selected to be used in the new prototype. Faulhaber 1628T with gearheads 66:1 and 134:1 and with an incremental encoder of 512 pulses per rotation are selected for the new prototype of Delta robot.

Selected motors satisfy the demands in terms of resolution for the precision necessity, speed, torque and also size to achieve a compact design. It can be seen from the table below that brushless DC motors are preferable over direct drive motors with their compact size and better parameters. The design is realized with the new motors considering the compactness and modularity as important issues and the third prototype of the Delta robot is developed.

	Resolution	Speed	Max. Torque
Planetary Gearhead 66:1	0.0026^o	16.4 rps	17.6 Ncm
Planetary Gearhead 134:1	0.0013^o	8 rps	34.84 Ncm
Direct Drive	0.36^o		3.5 Ncm

The final version of the Delta Robot and the 3D CAD model illustrating the compactness of the design is shown in Figure 6-32.

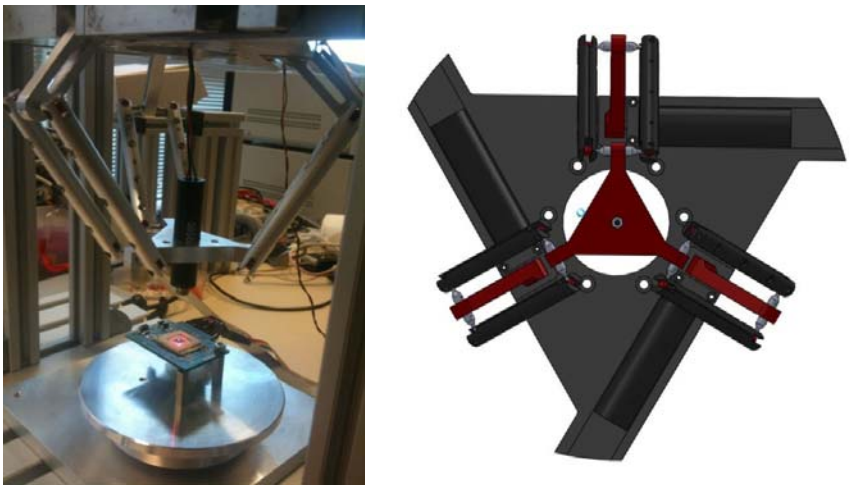


Figure 6-32 – (a) Third Prototype (b) Compactness of the design

First of all, in order to show the control performance, the reference is given directly to the motor as an angle input and the resulting behavior is shown in Figure 6-33. After achieving such a performance in the motor control then the mechanism control issue is considered and tested. The resolution of the encoders for the motors with planetary gearheads (134:1) is 0.0013^{o} and as it can be seen in the detailed figures below with different angle reference inputs, that the control at that resolution is achieved.

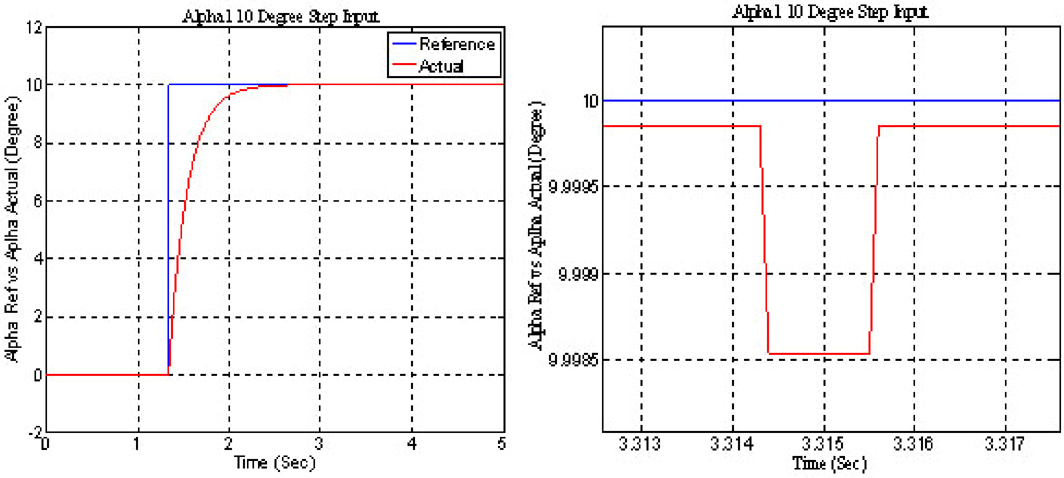


Figure 6-33 – (a) 10^{o} Angle Ref. vs. Actual (b) Detailed View

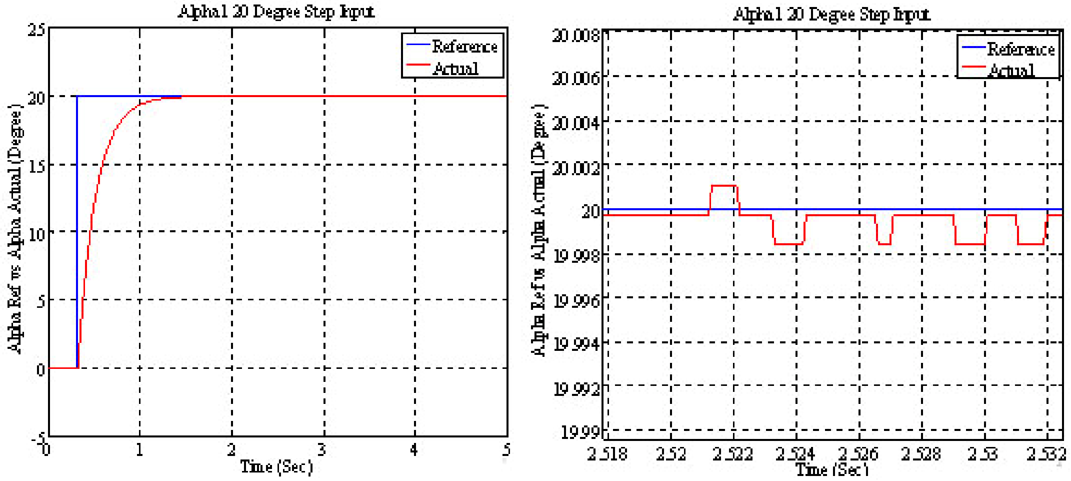


Figure 6-34 - (a) 20^{o} Angle Ref. vs. Actual (b) Detailed View

After testing the control performance with the direct angle reference to the motor, the forward kinematics is tested with reference to the X and Y input references to the Delta robot. The X and Y axes are given different input position references and the resulting outputs are shown in the following graphs. It should be noted that the motion starts with the start input after giving the reference input. That is the reason for the delay of the start of the motion seen in the figures below. X and Y axes with 10mm position reference inputs and the encoder outputs transformed into endeffector X position using forward kinematics equations are shown in Figure 6-35 and Figure 6-36. As it can be seen in the detailed view of the figures, the control provides 1 µm resolution in the task space calculated directly from the encoder outputs of the motors.

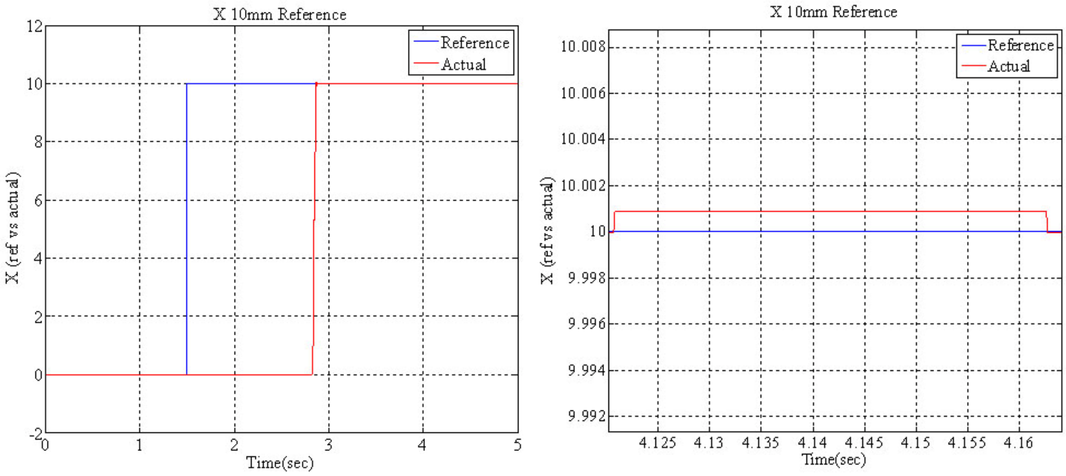


Figure 6-35 – (a) 10mm X position Ref. vs. Actual (b) Detailed View

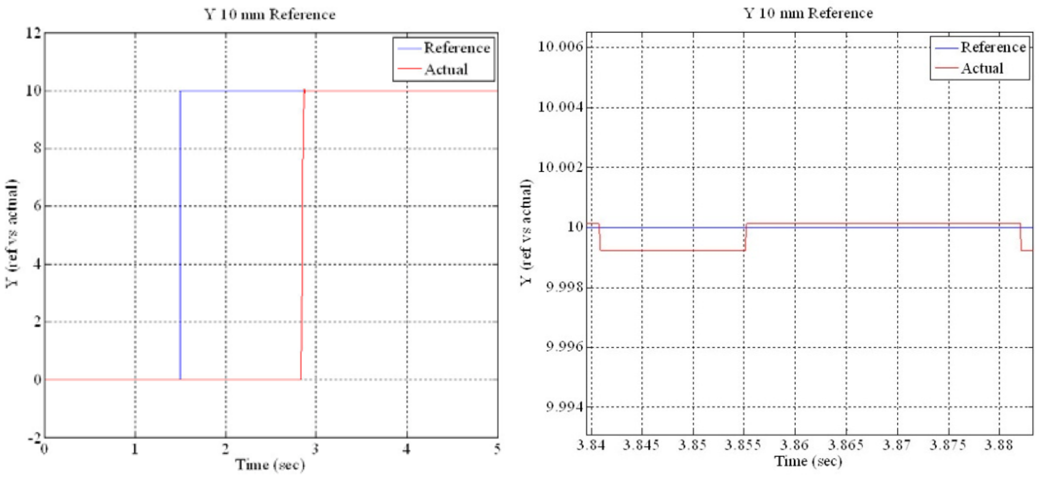


Figure 6-36 - (a) 10mm Y position Ref. vs. Actual (b) Detailed View

20 mm position reference inputs and the encoder outputs transformed into endeffector X and Y positions using forward kinematics equations are shown in Figure 6-36 and Figure 6-37 . The resolution corresponding to one encoder pulse achieved can be seen in the detailed figures and measured to be approximately 1 µm.

In order to test the real performance considering the impurities and intolerances of the Delta robot a task space measurement setup is built. The setup is configured first with the placement of the position sensor on a steady platform. However, that prevents the alignment of the sensor output and the endeffector of the robot so that the outputs cannot be displayed intersected to evaluate the performance. The first sensor measurement setup is given in Figure 6-32(a). For that reason, XY positioning sensor is located on a 3 DOF XYZ platform in order to align the sensor position with the position of the Delta robot. The positioning accuracy of the XYZ stages is 1 micrometer so that the sensor can be located precisely enough to provide the alignment. The position sensor is a laser actuated positioning sensor measuring the XY position of the laser source mounted to the end effector of the Delta robot. The sensor measurement has a limited area which is 4mm x 4mm. With a fixed placement of the sensor it is not possible to test the performance of the robot at any place in the workspace of the robot. This setup also enables the testing the performance of the system by moving the sensor to different locations within the workspace of the robot since the positioning stages have travel range of 15 mm. The enhanced measurement setup is shown in Figure 6-39.

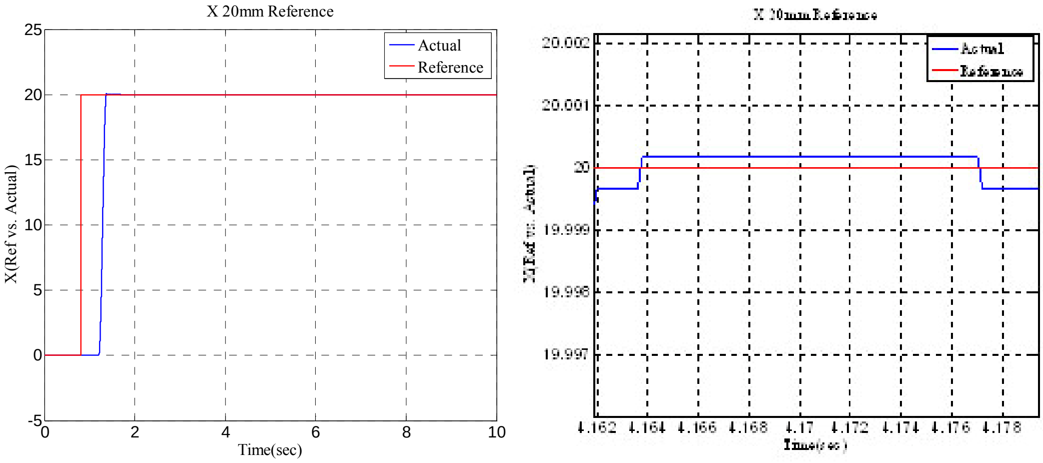


Figure 6-37 - (a) 20mm X position Ref. vs. Actual (b) Detailed View

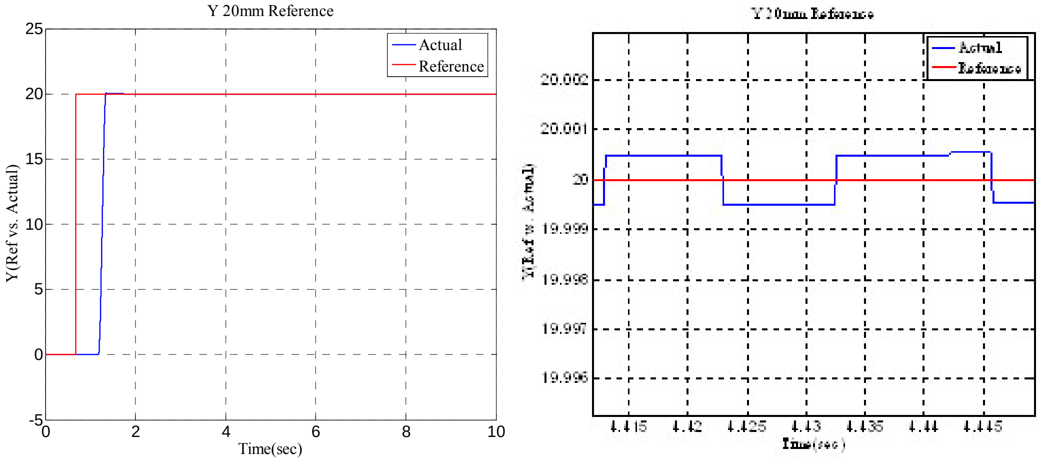


Figure 6-38 - (a) 20mm Y position Ref. vs. Actual (b) Detailed View

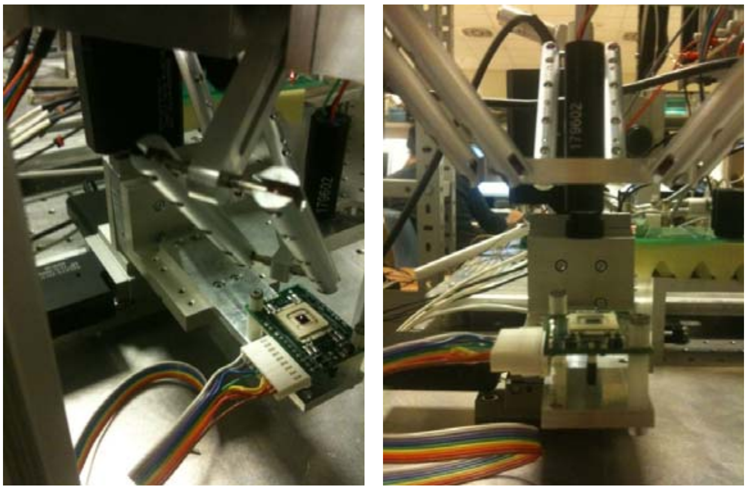


Figure 6-39 – Sensor Measurement Setup

Sinusoidal input references to X-Y axes of the Delta Robot are given with different amplitude and frequencies in order to achieve a circular reference. The following figures show the reference vs. encoder output and the reference vs. sensor output for a different radii circle input at different frequencies. The sensor outputs show slightly elliptic structures as a result of the horizontal alignment of the sensor and the Delta robot endeffector. That is because of the mounting of the sensor since it can not be perfectly aligned. Encoder outputs are giving the motor angles and using the forward kinematics equations the endeffector position is calculated and shown in the figures. However it is not representing the exact position of the end effector since the manufacturing and mounting imperfections of the robot can not be taken into account in such a calculation.

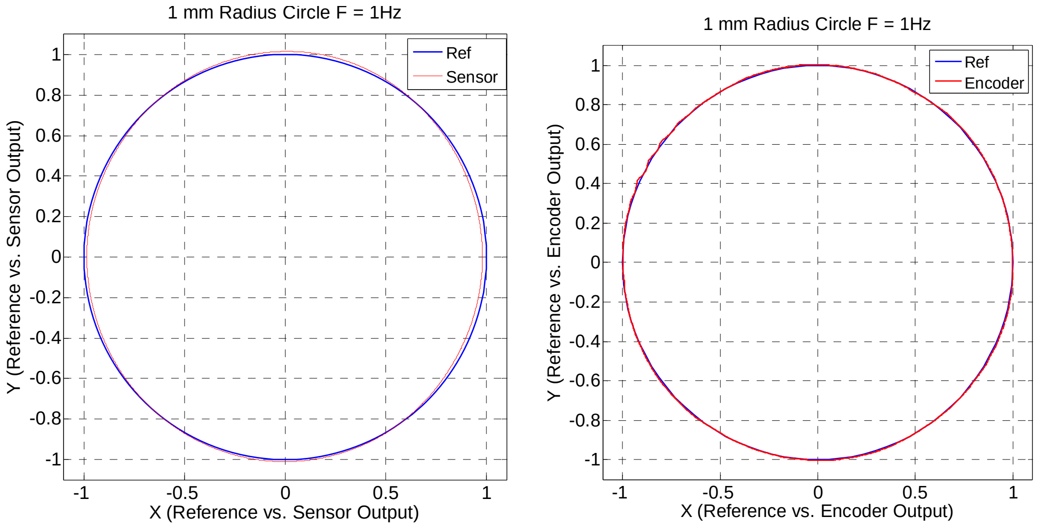


Figure 6-40 - 1mm Radius f=1 Hz Circle Reference (a) Ref. vs. Sensor (b) Ref. vs. Encoder

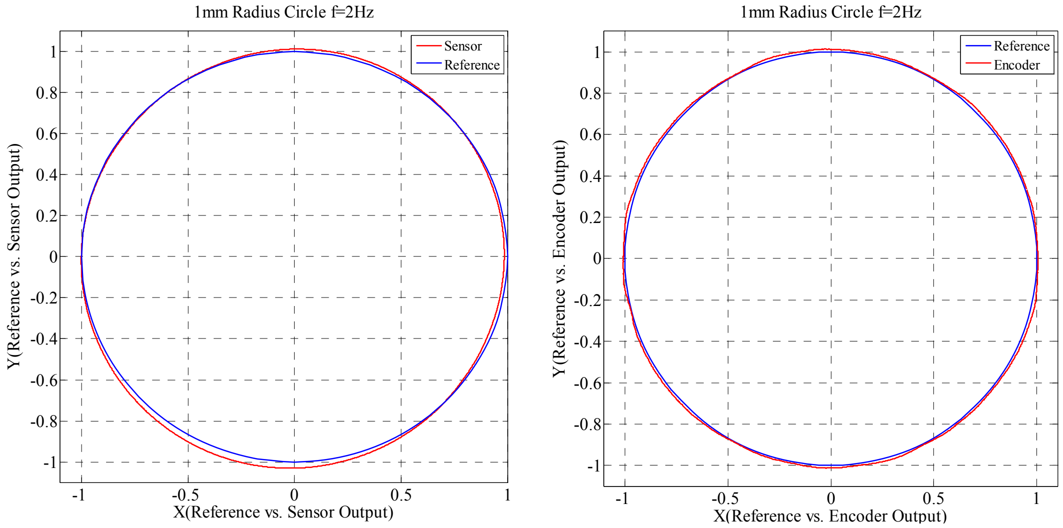


Figure 6-41 - 1mm Radius f=2 Hz Circle Reference (a) Ref. vs. Sensor (b) Ref. vs. Encoder

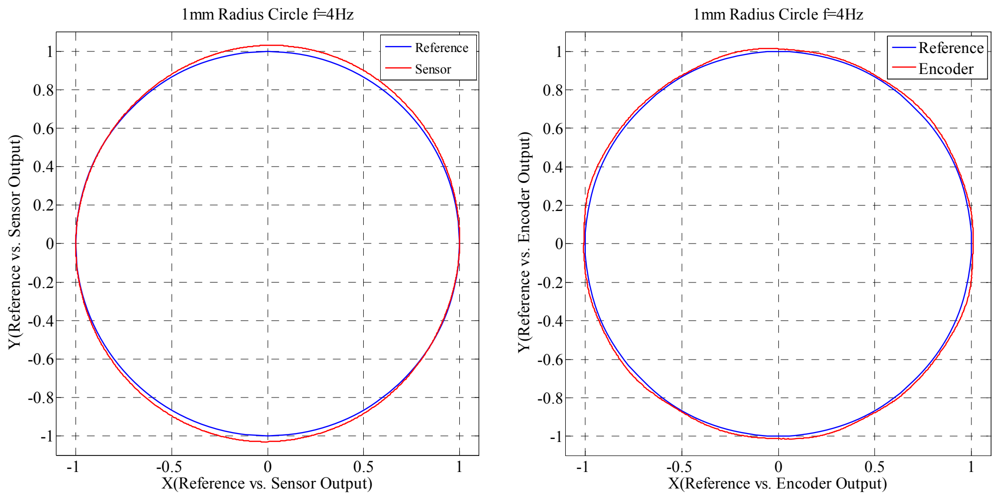


Figure 6-42 - 1mm Radius f=4 Hz Circle Reference (a) Ref. vs. Sensor (b) Ref. vs. Encoder

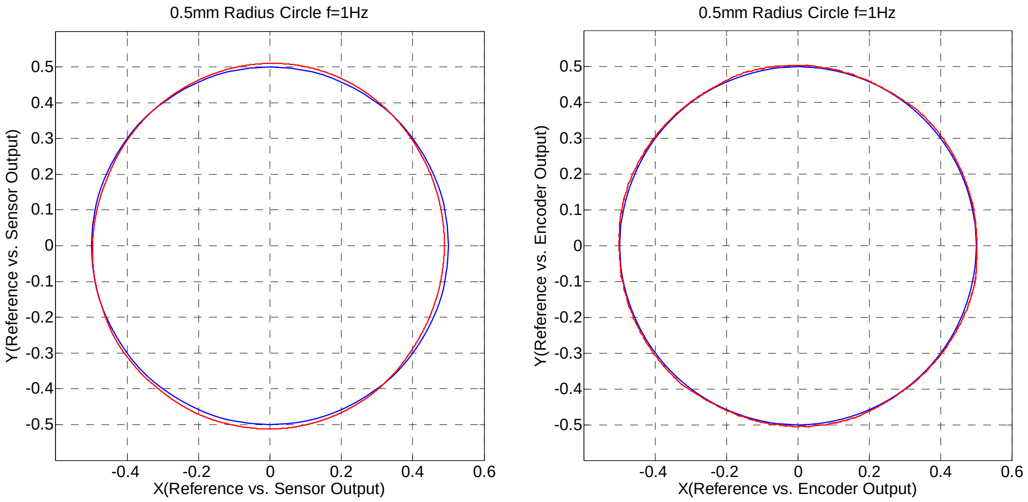


Figure 6-43 – 0.5mm Radius f=1 Hz Circle Reference (a) Ref. vs. Sensor (b) Ref. vs. Encoder

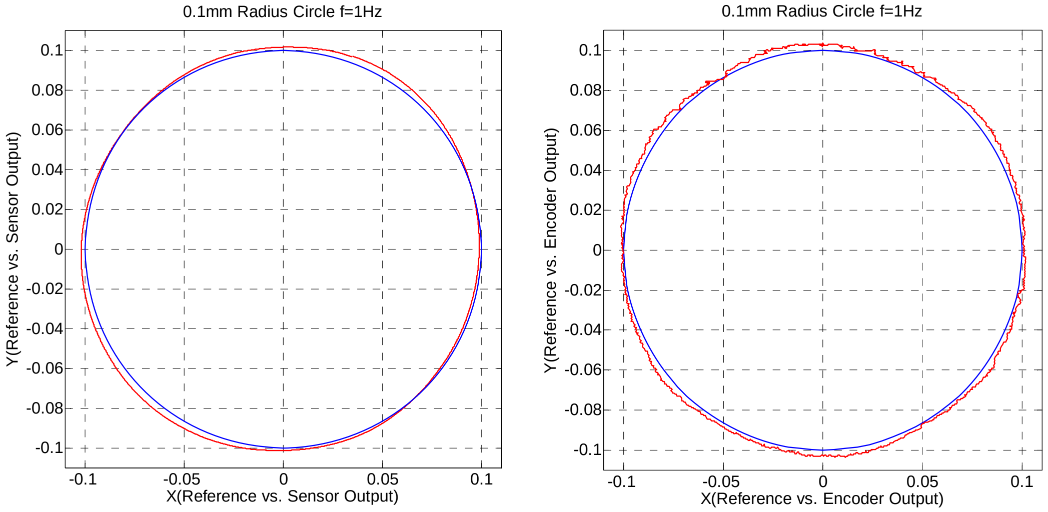


Figure 6-44 – 0.1mm Radius f=1 Hz Circle Reference (a) Ref. vs. Sensor (b) Ref. vs. Encoder

The performance analysis with respect to the data achieved from the position sensor is hard to make since there is that horizontal alignment problem. The read sensor data includes the error resulting from the control and the mechanical imperfections of the robot. In that sense using the elliptic output of the sensor, the angle between the Delta endeffector plane and the sensor plane cannot be perfectly calculated.

Figure 6-45 shows the detailed views of the reference circle and the sensor output for a 1mm radius circle reference. In order to calculate an approximate rotation angle between the Delta plane and the sensor plane, we need to assume that the Delta robot and the sensor center are perfectly aligned and neglect the errors resulting from the control and mechanical imperfections. As seen in the figures, the projection of the reference circle on the sensor plane forms an ellipse rotated in x and y axes since there are offsets in both x and y axes. The maximum error calculated from the following graphs is 15 µm which is even satisfactory for the design criteria targeted for the Delta robot.

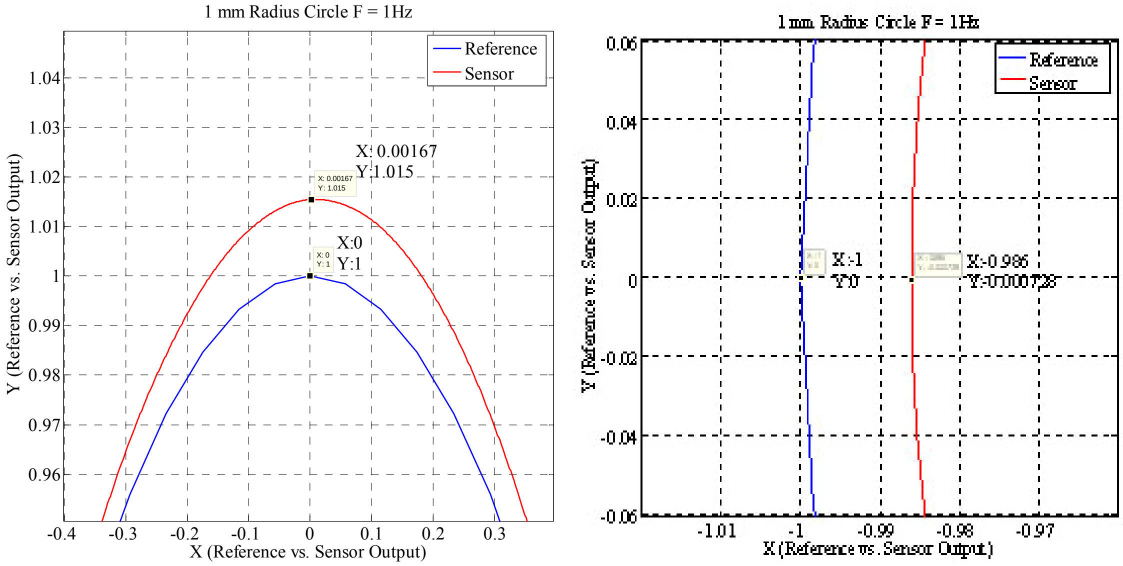


Figure 6-45 – Detailed view of the Reference vs. Sensor data for 1mm radius

The following figures are presented to demonstrate the task space motion and the joint space motion of the Delta robot together simultaneously. The reference of 2mm radius circle vs. robot’s actual endeffector position and the corresponding motor angle references vs. the actual motor angles are shown in Figure 6-46. The same position references with different frequencies are also shown in Figure 6-47 and Figure 6-48.

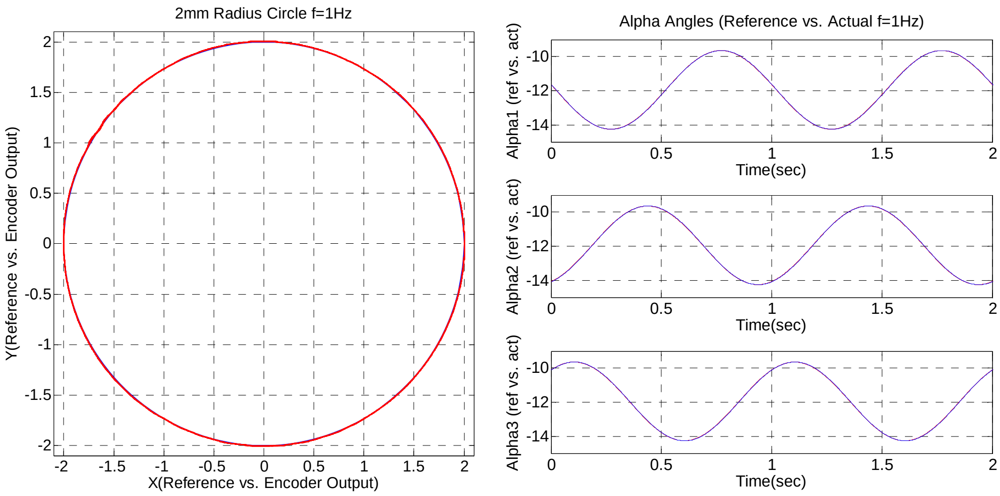


Figure 6-46 – (a) 2mm Circle Reference (f = 1Hz) (b) Corresponding motor angle ref. vs. actual pos.

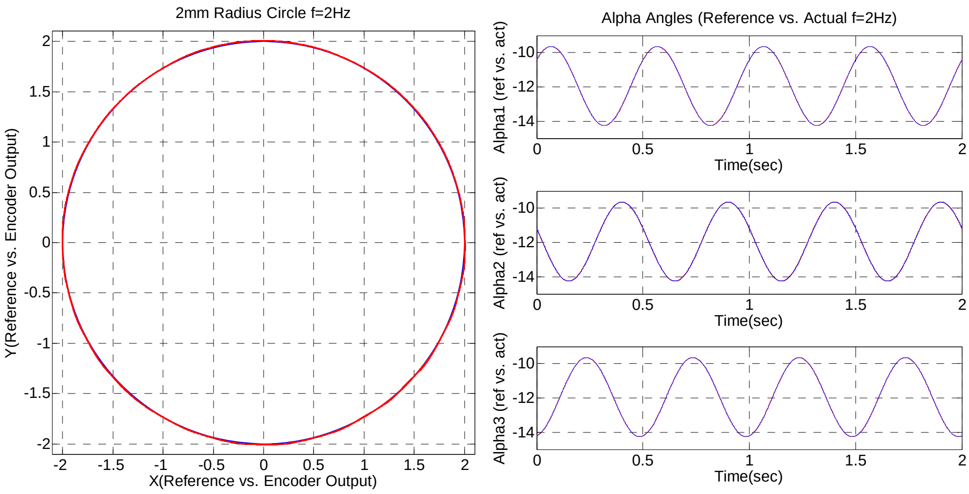


Figure 6-47 – (a) 2mm Circle Reference (f = 2Hz) (b) Corresponding motor angle ref. vs. actual pos.

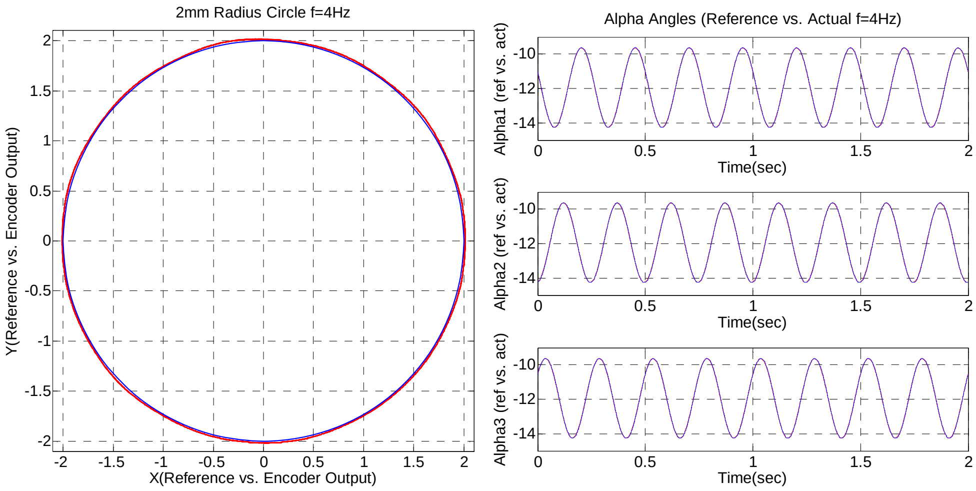


Figure 6-48 – (a) 2mm Circle Reference (f = 4Hz) (b) Corresponding motor angle ref. vs. actual pos.

6.2.2.3 Theory and Experiments

Several theories are examined and implemented in order to increase the performance of the Delta robot for the tasks to be implemented within the concept of the microfactory. The operational space formulation is summarized and expressed as in [51] and the application of the formulation to the Delta robot is given in the following section. Additional to operational space formulation, simulations on PD control with feedforward compensation are realized and results are given.

6.2.2.3.1 Operational Space Formulation

The dynamics and control of multibody systems has received much attention in several research areas both from the application and theoretical perspective. In the control of multibody systems, the most common method depends on the measurement of joint displacement which is called joint space control or the configuration space control. In a joint space control system, task specification is defined in terms of the robot's endeffectors motion, compliances, contact forces, etc. Then these terms are transformed into terms of joint positions, velocities, compliances, joint torques, etc. in order to provide the input to the control acting on the robot joints. This problem of transformation of terms in joint space control has been the motivation behind the work for the development of task space control methods. [46], [47], [48]

For the simplification of the control problem with the specification of a task space rather than the definition of the control task in the joint space, the operational space formulation was presented by Khatib [49], [50]. The joint space dynamics of a multibody system can be mapped into an appropriate task space using the operational space formulation which provides dynamic models for decoupled task and posture control in the task-level for the sake of control objectives.

The operational space formulation and control for the constrained dynamical systems with respect to the multiplier and minimization approaches is reviewed and the general formulation of constrained multibody systems into a task space using the operational space framework is introduced by Khatib in [51]. The framework offered allows the performing of a control task while synthesizing dynamic compensation for a multibody system which at the same time considers the system constraints.

The equations of motion for an unconstrained multibody system can be expressed as follows in terms of the joint space variables,

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau(t) \tag{6-18}$$

where q is the $n \times 1$vector of generalized coordinates, $M(q)$ is the $n \times n$ positive definite inertia matrix, $b(q,\dot{q})$ is the $n \times 1$ vector of coriolis and centripetal forces, $g(q)$ is the $n \times 1$ vector of gravity term and $\tau(t)$ is the $n \times 1$ vector of generalized torque vector.

In order to express the system dynamics in multiplier form, m constraint equations are introduced, $\emptyset(q) = 0$. Defining the gradient of Ø as the m × n constraint Jacobian matrix, Φ and introducing a set of constraint forces as a linear combination of the columns of $\boldsymbol{\Phi^T}$ multiplied with a vector of unknown Lagrange multipliers $\boldsymbol{\lambda}$ into equation (6-18) results in the dynamic equation,

$$\tau = M\ddot{q} + b + g - \boldsymbol{\Phi^T\lambda}\ (3) \tag{6-19}$$

which is subject to the constraint equations;

$$\emptyset(q) = 0, \qquad \frac{\partial\emptyset}{\partial q}\dot{q} = \Phi\dot{q}, \qquad \Phi\ddot{q} + \dot{\Phi}\dot{q} = 0 \tag{6-20}$$

Expressing the mass-weighted (right) inverse of Φ

$$\overline{\Phi} = M^{-1}\Phi^{\mathrm{T}}\left(\Phi M^{-1}\Phi^{\mathrm{T}}\right)^{-1} \tag{6-21}$$

where $\Phi\overline{\Phi} = 1$ and equivalently $\overline{\Phi}^{\mathrm{T}}\Phi^{\mathrm{T}} = 1$ and the $n \times n$ constraint null space matrix is defined as $\Gamma \triangleq 1 - \overline{\Phi}\Phi$.

For an alternate form of the constrained dynamical equations of motion, λ can be expressed as;

$$\lambda = -M_c[\Phi M^{-1}(\tau - b - g) + \dot{\Phi}\dot{\mathrm{q}}] \tag{6-22}$$

where M_c is the m × m constraint space mass matrix which reflects the system inertia projected at the constraint

$$M_c \triangleq (\Phi M^{-1}\Phi^{\mathrm{T}})^{-1} \tag{6-23}$$

Substituting into equation (6-19) we achieved,

$$M\ddot{q} + b + g = -\Phi^{\mathrm{T}} M_c \dot{\Phi} \dot{q} + \left(1 - \Phi^{\mathrm{T}} M_c \Phi M^{-1}\right)\tau + \Phi^{\mathrm{T}} M_c \Phi M^{-1}(b + g) \tag{6-24}$$

Defining the $\mathrm{m} \times 1$ vector of coriolis and centripetal forces projected at the constrained,

$$\alpha \triangleq M_c \Phi M^{-1} b - M_c \dot{\Phi} \dot{q} \tag{6-25}$$

And the $m \times 1$ vector of gravity forces projected at the constraint

$$\rho \triangleq M_c \Phi M^{-1} g \tag{6-26}$$

Also noting that;

$$\Gamma^T = 1 - \Phi^T \overline{\Phi}^T = 1 - \Phi^T M_c \Phi M^{-1} \tag{6-27}$$

Substituting the expressions the generalized constrained equation of motion is achieved;

$$\Gamma^T \tau = M\ddot{q} + b + g - \Phi^T(\alpha - \rho) \tag{6-28}$$

A set of task coordinates, x,can be related to the set of generalized coordinates,q, by

$$\dot{x} = J\dot{q} \tag{6-29}$$

Mapping the generalized constrained equation of motion into a task space via the dynamically consistent inverse of J yields;

$$(J^{\dagger})^T \Gamma^T \tau = M_t(q)\ddot{x} + b_t(q, \dot{q}) + g_t(q) + \gamma(q) \tag{6-30}$$

where

$$\begin{gathered} M_t(q) = (JM^{-1}J^T)^{-1} \\ b_t(q, \dot{q}) = (J^{\dagger})^T b - M_t \dot{J} \dot{q} \\ g_t(q) = (J^{\dagger})^T g \\ \gamma = -(J^{\dagger})^T \Phi^T(\alpha + \rho) \\ (J^{\dagger})^T = M_t J M^{-1} \end{gathered} \tag{6-31}$$

Since the actuation may not exist at all physical joints described by the generalized coordinates which is particularly the case of parallel mechanisms where many of the joints are passive. This can be resolved by using a selection matrix for the actuated joints, $S \in \mathbb{R}^{k \times n}$ and integrating into the equation;

$$(J^{\dagger})^{T}\Gamma^{T}S^{T}\tau_{k} = M_{t}(q)\ddot{x} + b_{t}(q,\dot{q}) + g_{t}(q) + \gamma(q) \tag{6-32}$$

where $\tau = S\tau_{k}$ and τ_{k} is the $k \times 1$ vector of generalized forces acting at the k actuated joints.

Modeling of the parallel Delta robot dynamics has been studied in the literature by using several methods. [52] and [42] used the Newton-Euler and Lagrange methods respectively, both treating the robot as a system of rigid bodies connected by frictionless kinematic pairs. [43] and [53] used a method based on the direct application of the Hamilton's principle to solve the inverse dynamics, latter implementing for real time application in the control law of the direct-drive version of the Delta robot. [54] proposed a dynamic model based on the virtual work principle and giving the mass matrix of the robot evaluated based on kinetic energy considerations. In [55] and [56], a modeling approach, the goal of which is the derivation of fast models by defining an optimal set of parameters in order to simplify the equations, is proposed. [57] proposed a method, also based on virtual work principle, for the derivation of the dynamic equation in an explicit linear function of the dynamic parameters.

For further simulations and the experiments with the Delta robot in order to implement the algorithms, the dynamics of the Delta robot is modeled using Autolev, a symbolic manipulation software tool useful for generating equations of motion for mechanical systems. The key components in dynamics like velocities, accelerations and angular moments are vector quantities. Vector algebra and vector calculus should be employed to deal with these quantities analytically. Autolev provides easy handling of vector algebra and calculus with proper definition of the vector and scalar quantities with respect to the defined reference frames. The modeling of the dynamics of Delta robot in Autolev using Kane's method [58] is defined in the following parts of this section. Further details for the implementation of the dynamics and the functional block library established and used in the simulations are given in the Appendix.

The Newtonian frame of reference, $N,$ is defined in the middle of the upper base of the Delta robot since the robot is fixed from that part. The first arm of the Delta robot is defined according to the Newtonian frame. The other two reference frames, P and R, defined with orientation of 120$^{\text{o}}$ and 240$^{\text{o}}$ degrees respectively at the base are for the definition of the second and the third arms of the robot. There are seven bodies defined for the Delta robot; two for each arm and one for the nacelle. Each body is assigned its own reference frame, however, for the lower arm since there is motion in two planes another reference frame is assigned to define the second motion of the parallelogram structure. In order to define the position vectors, origin, endeffector position and necessary points for each arm are defined for the shoulder, elbow and the wrist. The bodies, reference frames and points are shown in Figure 6-49.

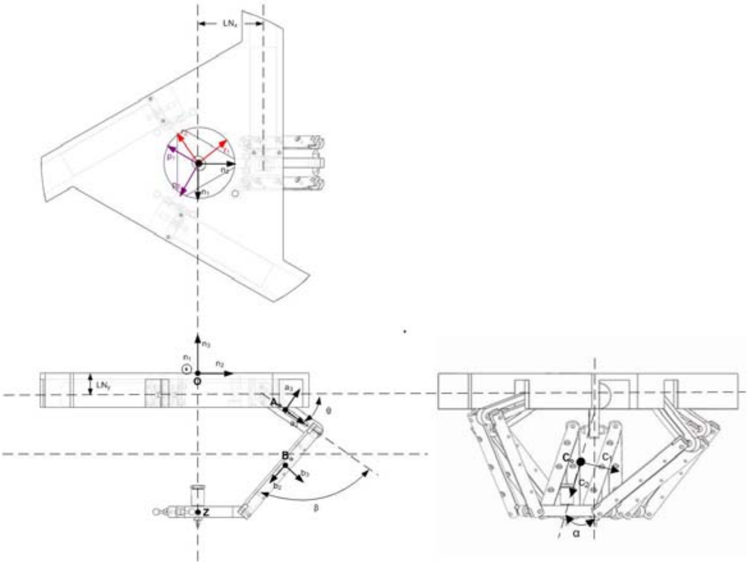
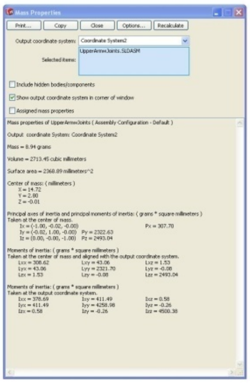


Figure 6-49 – Bodies, reference frames and points

The Delta robot kinematic parameters are depicted in   Figure 6-9 and defined accordingly for the AUTOLEV code. The mass and inertia values for each body are defined and the values are calculated in Solidworks CAD software with the proper assignment of the reference frames. Mass and inertia calculations of the bodies are shown in Figure 6-50, Figure 6-51 and Figure 6-52 respectively.



Output coordinate system: Coordinate System2

Selected items: UpperArmwJoints.SLDASM

□ Include hidden bodies/components

☑ Show output coordinate system in corner of window

□ Assigned mass properties

```
Mass properties of UpperArmwJoints ( Assembly Configuration - Default )

Output  coordinate System: Coordinate System2

Mass = 8.94 grams

Volume = 2713.45 cubic millimeters

Surface area = 2368.89 millimeters^2

Center of mass: ( millimeters )
	X = 14.72
	Y = 2.80
	Z = -0.01

Principal axes of inertia and principal moments of inertia: ( grams * square millimeters )
Taken at the center of mass.
	 Ix = (-1.00, -0.02, -0.00)                    Px = 307.70
	 Iy = (-0.02, 1.00, -0.00)   Py = 2322.63
	 Iz = (0.00, -0.00, -1.00)   Pz = 2493.04

Moments of inertia: ( grams * square millimeters )
Taken at the center of mass and aligned with the output coordinate system.
	Lxx = 308.62            Lxy = 43.06             Lxz = 1.53
	Lyx = 43.06             Lyy = 2321.70           Lyz = -0.08
	Lzx = 1.53              Lzy = -0.08             Lzz = 2493.04

Moments of inertia: ( grams * square millimeters )
Taken at the output coordinate system.
	Ixx = 378.69            Ixy = 411.49            Ixz = 0.58
	Iyx = 411.49            Iyy = 4258.98           Iyz = -0.26
	Izx = 0.58              Izy = -0.26             Izz = 4500.38
```

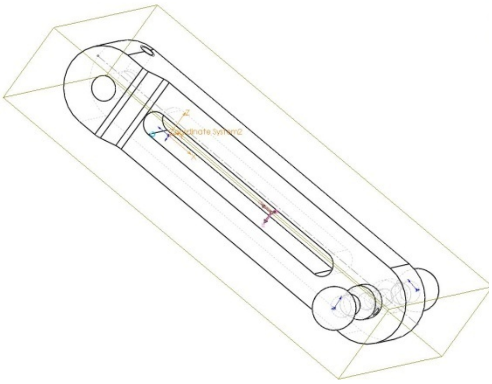


Figure 6-50 – Upper Arm Mass/Inertia Values

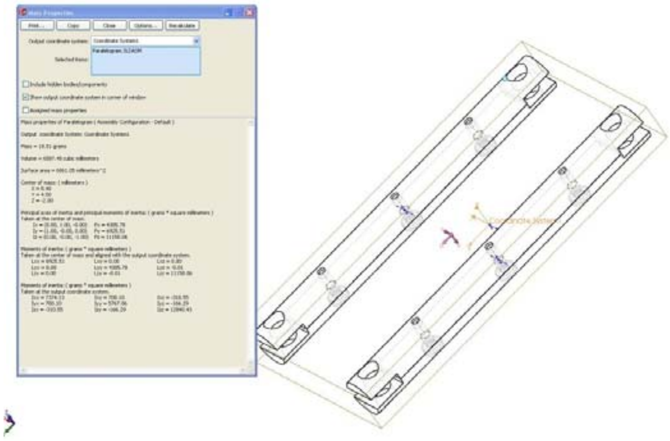


Figure 6-51 – Lower Arm Mass/Inertia Values

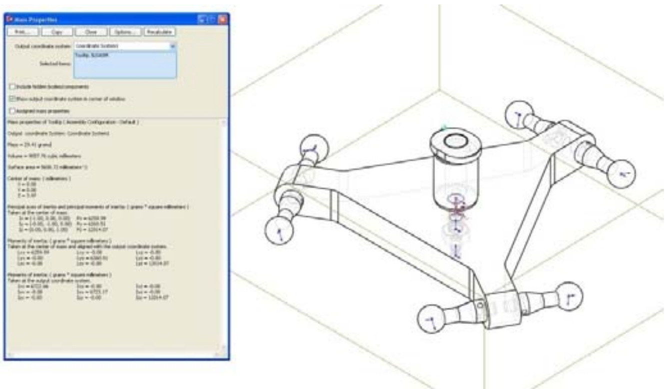


Figure 6-52 – Nacelle Mass/Inertia Values

The relative orientations between the bodies are defined as simple rotations or direction cosines. With these definitions, each frame can be defined in terms of each other and the Newtonian frame. The upper arm motion actuated by the motors is defined by a simple rotation between the fixed frames defined at the base plate and the actuated upper arm, the rotation of which is denoted by input angle α_i $(i = 1,2,3)$. The relative orientation of the upper arm and the lower parallelogram is defined using a direction cosine by two angles β_i and γ_i $(i = 1,2,3)$. With the definition of the relative orientations of the bodies and frames, each frame can be defined with respect to each other or the Newtonian frame which allows further mathematical operations to be realized.

P^{OSh} denotes the position vector from O to Sh (from origin to shoulder). Using this notation, position vectors for each arm configuration are defined as follows;

$$\begin{aligned} P^{OSh_i} &= R_a\overrightarrow{n_{\iota 2}} \quad (i = 1,2,3) \\ P^{Sh_iEl_i} &= L_a\overrightarrow{a_{\iota 2}} \quad (i = 1,2,3) \\ P^{El_iWr_i} &= L_b\overrightarrow{b_{\iota 2}} \quad (i = 1,2,3) \\ P^{Wr_iZ} &= -R_b\overrightarrow{n_{\iota 2}} \quad (i = 1,2,3) \\ P^{OZ} &= x\overrightarrow{n_\iota} + y\overrightarrow{n_2} + z\overrightarrow{n_3} \end{aligned} \tag{6-33}$$

The angular velocity of a rigid body A with mutually perpendicular unit vectors A_1, A_2 and A_3 moving in the reference frame N, ${}^{N}\omega^{A}$is defined as

$${}^{N}\omega^{A} \triangleq A_1 \frac{{}^{N}dA_2}{dt} \cdot A_3 + A_2 \frac{{}^{N}dA_3}{dt} \cdot A_1 + A_3 \frac{{}^{N}dA_1}{dt} \cdot A_2 \tag{6-34}$$

The angular acceleration,${}^{N}\alpha^{A}$, is defined as the first time derivative of the angular velocity, ${}^{N}\omega^{A}$, in N.

$${}^{N}\alpha^{A} \triangleq \frac{{}^{N}d\,{}^{N}\omega^{A}}{dt} \tag{6-35}$$

P^{OZ} denotes the position vector from the point O in the reference frame N to a point Z moving in N. The velocity of Z in N, ${}^{N}v^{Z}$ is defined as

$${}^{N}v^{Z} \triangleq \frac{{}^{N}dP^{OZ}}{dt} \tag{6-36}$$

And the acceleration, ${}^{N}a^{Z}$is defined as

$${}^{N}a^{Z} \triangleq \frac{{}^{N}d\,{}^{N}v^{Z}}{dt} \tag{6-37}$$

The gravitational force acting on the system where g denotes the gravitational constant;

$$G = -gN_3 \tag{6-38}$$

The external forces acting on the endeffector;

$$F_{ex} = F_xN_1 + F_yN_2 + F_zN_3 \tag{6-39}$$

The torques provided by the actuators for three links;

$$\begin{aligned} T_A &= -TO_AA_1 \\ T_D &= -TO_DD_1 \\ T_G &= -TO_GG_1 \end{aligned} \tag{6-40}$$

In order to derive the equation of motion, recalling the definition of Lagrangian function as the difference between the kinetic and potential energies;

$$\mathcal{L} = T - V \tag{6-41}$$

The Lagrangian equation;

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = 0 \tag{6-42}$$

results in the differential equations that describe the equations of motion of the system. Since the potential energy is only a function of position, Lagrange's equations can be defined as follows;

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_i}\right) - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} \tag{6-43}$$

The generalized coordinates for the system are defined to be;

$$q_i = \{\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \gamma_3, x, y, z\}$$

The generalized forces defined as the applied forces in the generalized coordinates;

$$Q_i = \sum_{i=1}^{n} F_i \cdot \frac{\partial r_i}{\partial q_i} \tag{6-44}$$

where F_i are the applied forces to each body and r_i are the position vectors to the center of mass of each body from the origin.

The declaration of the equations of motion for the unconstrained system is the expressed as follows;

$$EoM = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i}\right) - \frac{\partial \mathcal{L}}{\partial q_i} - Q_i = 0 \tag{6-45}$$

For the definition of the constrained equations of the motion, the two linearly independent loop closures are defined as,

$$\begin{aligned} L_1(q) &= P^{OSh_1} + P^{Sh_1El_1} + P^{El_1Wr_1} + P^{Wr_1Z} - P^{Wr_2Z} - P^{El_2Wr_2} - P^{Sh_2El_2} \\ &\quad - P^{OSh_2} \\ L_2(q) &= P^{OSh_3} + P^{Sh_3El_3} + P^{El_3Wr_3} + P^{Wr_3Z} + P^{OZ} \end{aligned} \tag{6-46}$$

The six configuration constraint equations are derived to be;

$$\begin{gathered}\emptyset_i(q) = L_1(q) \cdot N_i \;\; (i = 1,2,3)\\ \emptyset_{i+3}(q) = L_2(q) \cdot N_i \;\; (i = 1,2,3)\end{gathered} \tag{6-47}$$

The constraint Jacobian is defined as the partial derivative of the configuration constraints in terms of the generalized coordinates;

$$\Phi(j,i) = \frac{\partial \emptyset_j(q)}{\partial q_i} \quad (i = 1,..,12, j = 1,..,6) \tag{6-48}$$

Defining the task to control the active elbow joints; that is $x \triangleq (q_{10} \quad q_{11} \quad q_{12})$, the task Jacobian has the form;

$$J = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Due to the passive nature of all other joints the component of active force, f_C, acting along the constraint direction is chosen to be zero.

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Defining all the terms and deriving the constraint equations of motion for the Delta robot, the operational space formulation can be applied as depicted in equation (6-32). With the selection of the task Jacobian, the task space can be formulated in different configurations. All the components of the equations of motion of the system are obtained using the Autolev solutions to be used in the Simulink simulation environment. First the model of the operational space is formed for the task is defined to control the generalized coordinates q_1, q_2 and q_3. The task Jacobian is defined as;

$$J = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

After establishing the system, the simulation is run and the results are shown in the Figure 6-53.

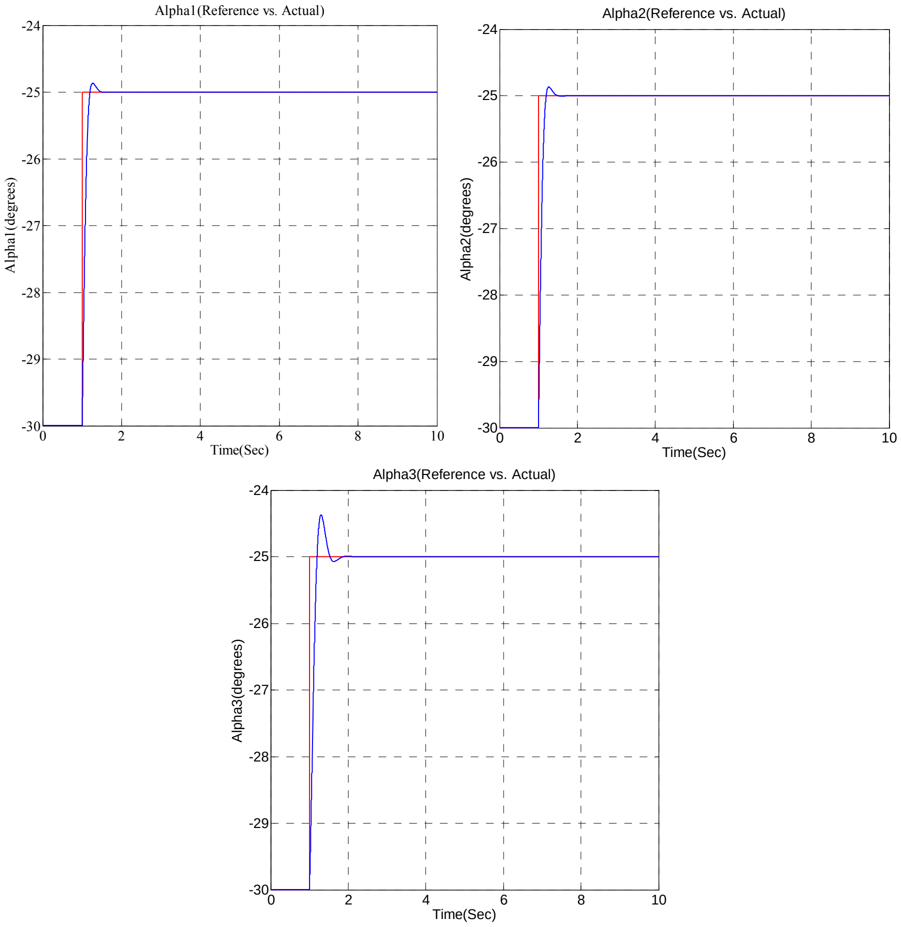


Figure 6-53 – Alpha1-2-3 Ref. vs. Actual (Operational Space Formulation)

6.2.2.3.2 PD Control with Feedforward Compensation Simulations

Recalling the equation (6-18) for Delta robot's dynamical equation of motion in the configuration space

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau(t) \tag{6-49}$$

The motion control in joint space primarily targets to achieve the robot joint position q to track the desired joint position q_d. $\tilde{q} = q_d - q$ denoting the joint position error, the control objective is try to provide that

$$\lim_{t\to\infty} \tilde{q} = 0 \tag{6-50}$$

The PD control law with feedforward compensation can be written as the combination of a linear PD feedback and a feedforward computation of the nominal robot dynamics [59] as;

$$\tau = K_P\,\tilde{q} + K_D\dot{\tilde{q}} + M(q_d)\,\ddot{q_d} + b(q_d,\dot{q_d})\,\dot{q_d} + g(q_d) \tag{6-51}$$

where K_P and K_D are $n \times n$ diagonal positive definite proportional and derivative gain matrices.

In order to test the performance of the system, the simulations are first realized without feedforward compensation to be able to compare the results. 5 mm radius circle reference is given with the phase shifted sinusoidal inputs to the x and y axes of the robot. The PD control of Delta robot and the results are shown in the following figures containing the circle reference vs. actual and the results for the three joint positions;

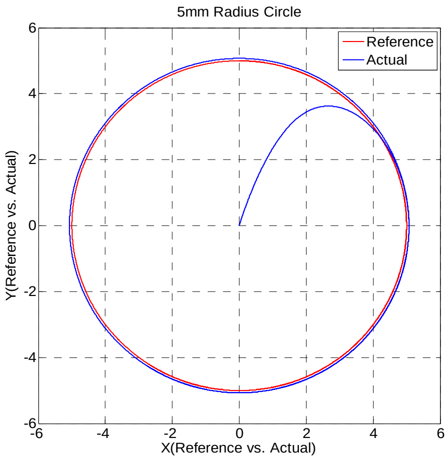


Figure 6-54 – 5mm radius circle reference vs. actual (PD Control)

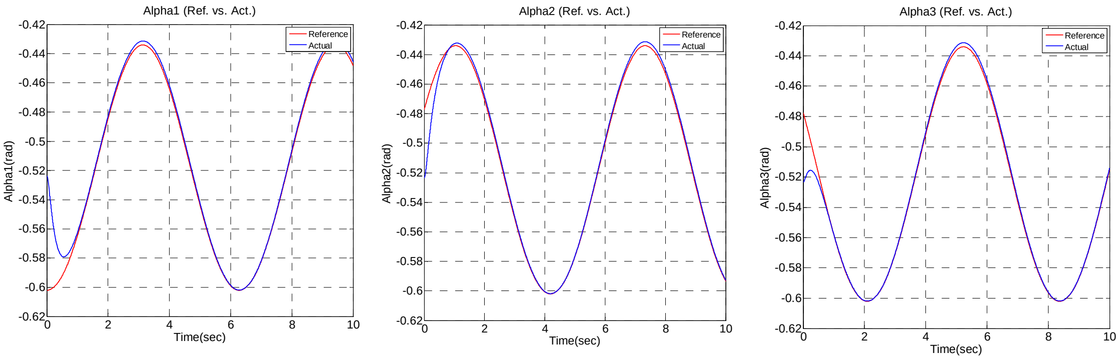


Figure 6-55 – Alpha1-2-3 Ref. vs. Actual (PD Control)

The PD control with gravity compensation block diagram is shown in Figure 6-56. The PD control with gravity compensation consists of a linear PD feedback plus a gravity computation.

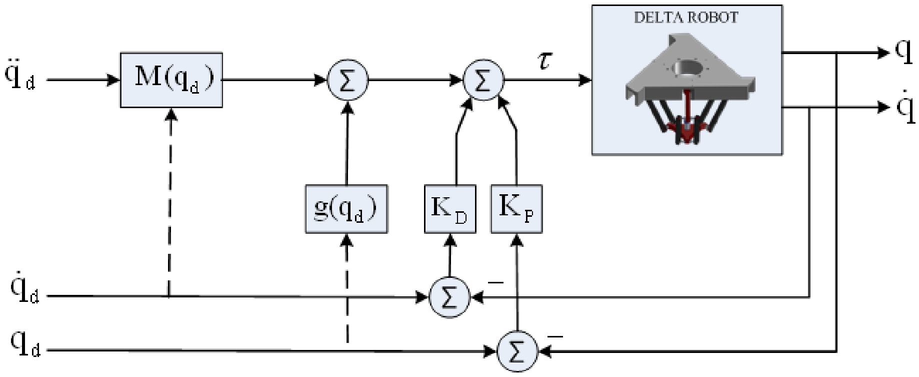


Figure 6-56 – PD Control with Gravity Compensation

The PD control parameters are kept the same in order to compare the performance of the feedforward compensation. The results are shown in the following figures.

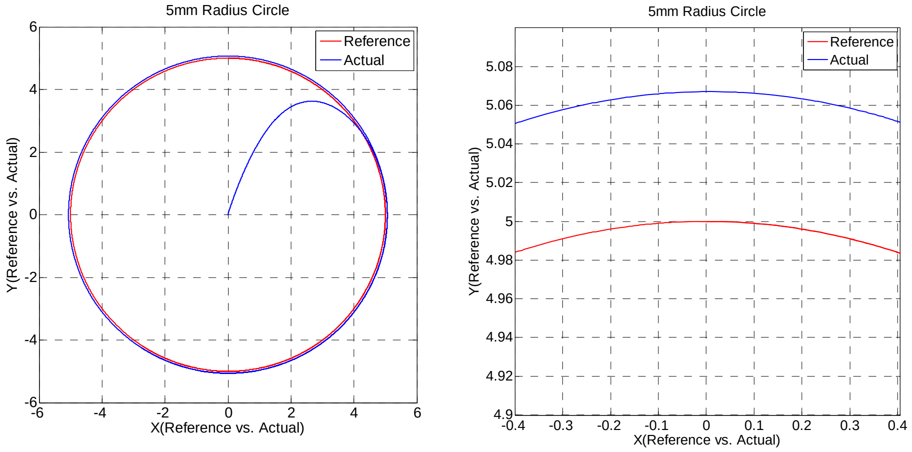


Figure 6-57 – a) 5mm radius circle ref. vs. act. (PD Control with gravity compensation) b) Detailed View

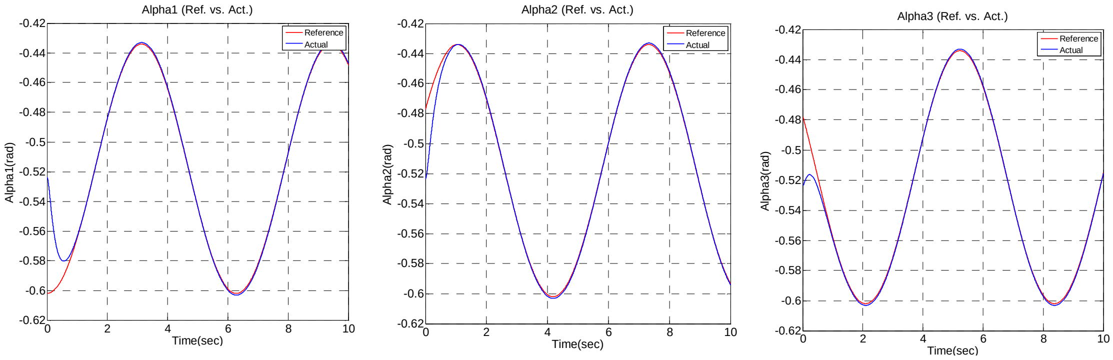


Figure 6-58 – Alpha1-2-3 Ref. vs. Actual (PD Control with gravity compensation)

The PD control with feedforward compensation schematics is shown in Figure 6-59. The PD control with feedforward compensation consists of a linear PD feedback plus a feedforward computation of the nominal robot dynamics.

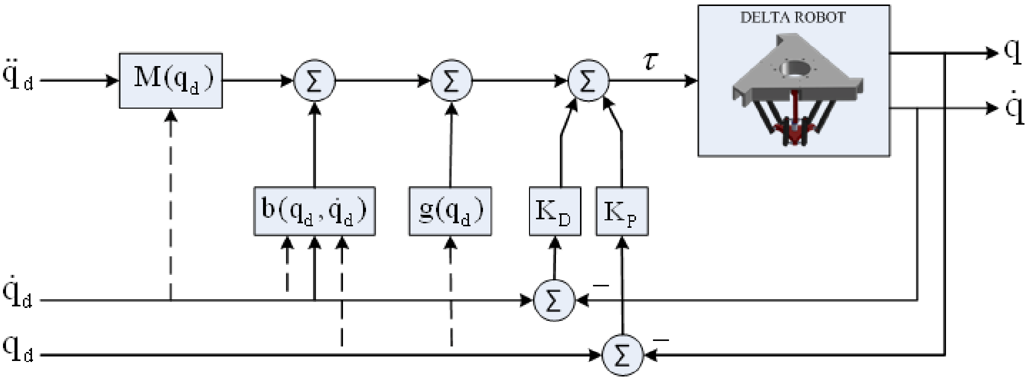


Figure 6-59 - PD Control with Feedforward Compensation

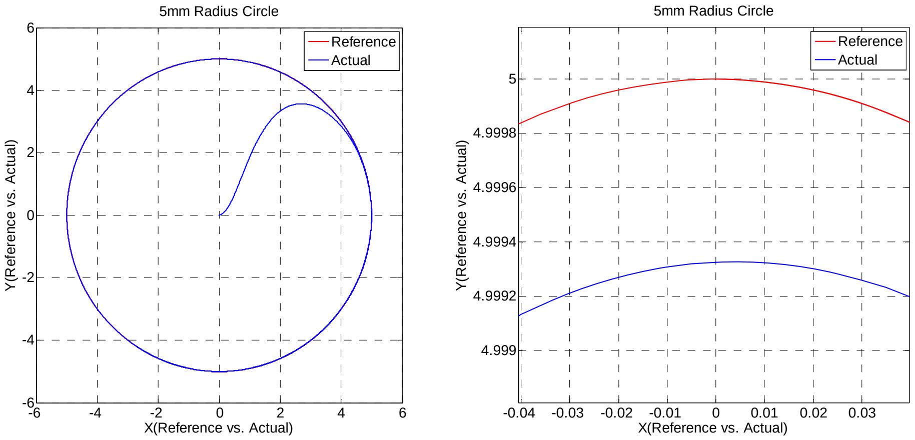


Figure 6-60 – a) 5mm radius circle ref. vs. act. (PD Control with feedforward compensation) b) Detailed View

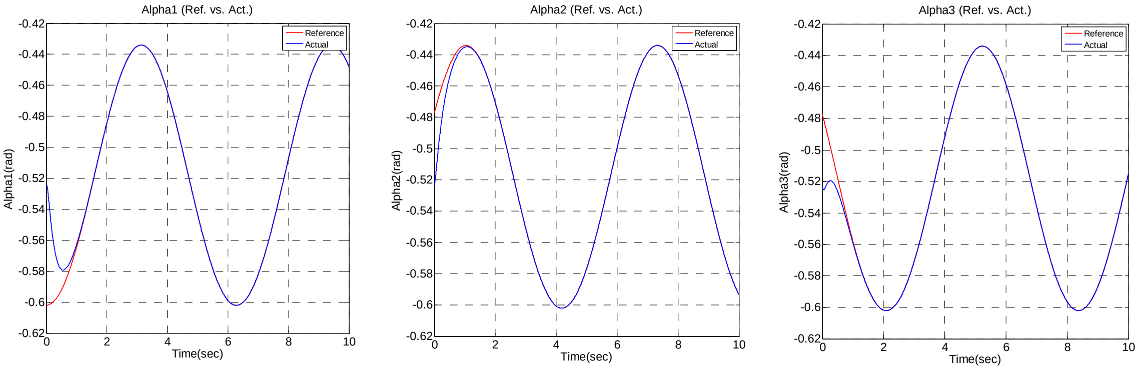


Figure 6-61 – Alpha1-2-3 Ref. vs. Actual (PD Control with gravity compensation)

As it can be seen from the experimental results above, addition of the feedforward computation of the nominal robot dynamics increases the control performance. It should be noted that the results shown are simulation results. The simulations are realized using the functional block library for the Delta robot given in Appendix.

6.2.3 Pantograph

In addition to the miniaturized Delta robot, another parallel robot, pantograph, is designed and manufactured to be used as a manipulator in the system. Each design step realized for the Delta robot is realized also for the pantograph mechanism and three prototypes are developed. Pantograph is a 5 links parallel mechanism with 2 degrees of freedom moving in X and Y coordinate planes. The Pantograph device is first introduced by Ramstein and Hayward [60] in 1994 in order to develop a haptic interface which measures position and velocity of a manipulated knob and displays forces in two dimensions over a wide frequency range. The main area of study for the pantograph as a manipulator is micro assembly applications in which accuracy requirements are very high; therefore, precision and since repeatability for such assembly systems must be in the micron to nanometer range for automatic assembly of structures with very small size (millimeter to micron). The pantograph mechanism developed as the final prototype in order to show the structure is given in Figure 6-62.

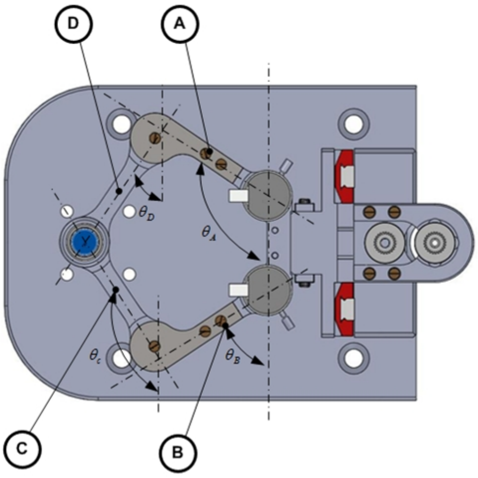


Figure 6-62 – Pantograph Mechanism

6.2.3.1 Kinematics of Pantograph

The kinematic structure is a five-bar planar linkage represented in Figure 6-63. The end-plate is located at point P_3 and moves in a plane with two degrees-of-freedom with respect to the ground link, where the actuators and sensors are located at P_1 and P_5. The configuration of the device is determined by the position of the two angles q_1 and q_4 and the force at the tool tip P_3 is due to torques applied at joints 1 and 5.

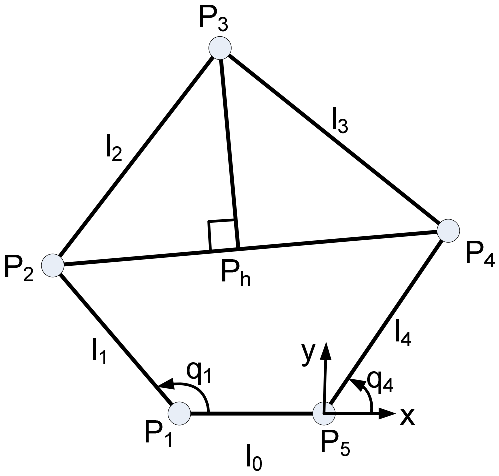


Figure 6-63 – Kinematic Model of the Pantograph

The forward kinematics problem consists of finding the position of point P_3 from the two actuated joint angles q_1 and q_4. The base frame is fixed and the reference frame is located at point P_5. The kinematics of the pantograph is solved using various approaches, and the one presented in [61] is implemented for the pantograph. All approaches share the observation that P_3 is at the intersection of two circles center and radius of each are known. The circles with radii l_2 and l_3 are centered at points P_2 and P_4which are calculated as;

$$P_4(x_4, y_4) = [l_4 cos(\mathrm{q}_4), l_4 sin(q_4)]^T \qquad (6\text{-}52)$$
$$P_2(x_2, y_2) = [\,\mathrm{l}_1 cos(q_1) - \mathrm{l}_0, l_1 sin(\mathrm{q}_1)]^T$$

Defining $P_h = (x_h, y_h)$ as the intersection between the segment P_2P_4 and the height of triangle $P_2P_3P_4$ as shown in Figure 6-63;

$$\|P_4 - P_h\| = \frac{(l_3^2 - l_2^2 + \|P_2 - P_4\|^2)}{(2\|P_2 - P_4\|^2)} \qquad (6\text{-}53)$$

$$P_h = P_4 + \frac{\|P_4 - P_h\|}{\|P_4 - P_2\|}(P_2 - P_4) \qquad (6\text{-}54)$$

$$\|P_3 - P_h\| = \sqrt{l_3^2 - \|P_4 - P_h\|^2} \qquad (6\text{-}55)$$

The position of the end effector $P_3 = (x_3, y_3)$ can be calculated as;

$$x_3 = x_h \pm \frac{\|P_3 - P_h\|}{\|P_4 - P_2\|}(y_2 - y_4) \qquad (6\text{-}56)$$

$$y_3 = y_h \mp \frac{\|P_3 - P_h\|}{\|P_4 - P_2\|}(x_2 - x_4)$$

The inverse kinematics solution of the parallel mechanisms is easier than the forward kinematics solutions. The inverse kinematics of the pantograph aims to find the actuated joint angles q_1 and q_4 given the position of end effector point P_3. P_2 and $P_4$are two passive joints and the end effector position cannot be defined directly using these passive joints since the values cannot be measured. Therefore, the pantograph is divided into three triangles, as shown in Figure 6-64, and the end point position can be calculated using the cosines theorem.

$$\begin{gathered}\alpha_4 = \arccos\left(\frac{l_4^2 - l_3^2 + \|P_5,P_3\|}{2l_4\sqrt{\|P_5,P_3\|}}\right)\\ \beta_4 = atan2(y_3, -x_3)\\ \beta_1 = \arccos\left(\frac{l_1^2 - l_2^2 + \|P_1,P_3\|}{2l_1\sqrt{\|P_1,P_3\|}}\right)\\ \alpha_1 = atan2(y_3, x_3 + l_0)\end{gathered} \tag{6-57}$$

$$q_4 = \pi - \alpha_4 - \beta_4,\ \ q_5 = \alpha_1 + \beta_1 \tag{6-58}$$

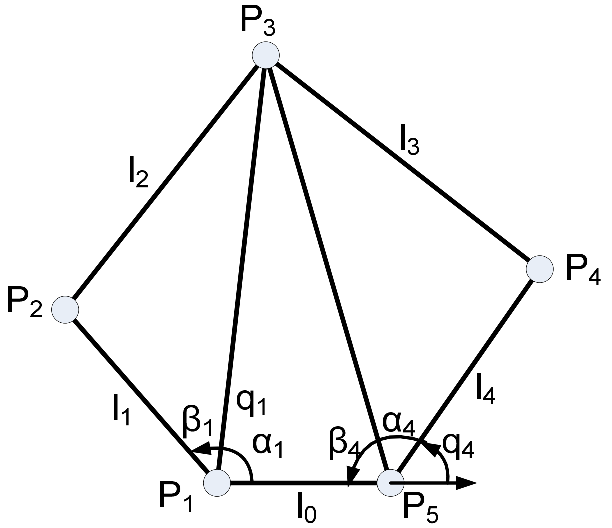


Figure 6-64 – Dividing the pentagon into three triangles

The link lengths of the designed pantograph are determined to be $l_1 = l_2 = l_3 = l_4 = 40$ mm and $l_0 = 30$ mm. Initial position of the pantograph is at $q_4 = 60^o$ and

q_1= 120^o and the offset from the reference frame at the initial position is calculated as $x_{off} = -15$ mm and $y_{off} = 54$ mm and the kinematic calculations are done accordingly.

6.2.3.2 Prototypes

The first prototype of the pantograph is designed as a sample holder XYR stage which allows backlight illumination with the Ø20 mm opening at the end effector. It is desired to work under an optical microscope holding the sample holder unit on which the manipulation operations can be realized. As a result of the opening, arms of the pantograph have to be thick since the bearing used at the end effectors the design parameters of the mechanism. In that prototype, manufacturing of the arms and other parts are realized using a rapid prototyping machine. The important aspect of the pantograph is its planar parallel mechanism which is the nature of its interface. At this interface, the tangential forces may cause deformations at fingertips [61].

Necessary degrees of freedom for the platform are three; two translational axes to allow the work piece to be positioned in X and Y orthogonal axes (Z axis can be added) and an independent rotational axis in order to orientate the work piece. For the orientation, an integrated rotational axis at the end tip is added to the design.

The stage should span an area of 20mm x 20mm which is determined to be the travel range of the actuator. So that the link lengths are determined in order to provide that travel range. In the following figure link lengths and the workspace is shown with the dexterous workspace of 20mm x 20mm square.

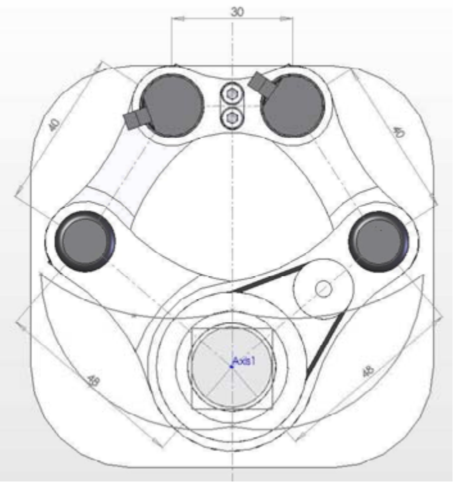


Figure 6-65 – Link Lengths and Workspace

In order to eliminate or decrease the bending through the end tip of the actuator, link design, junctions and motor-shaft couplings should be carefully considered. The link design is realized as I beam structure which is a very efficient form for the compensation of bending as well as shear according to the Euler-Bernoulli beam equation.

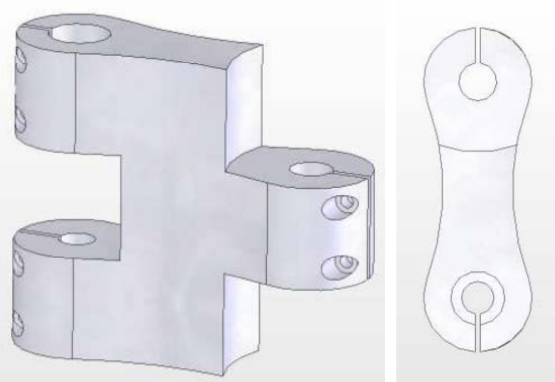


Figure 6-66 – Link Design

The sectional profiles of the arms are optimized in order to minimize weight without sacrificing stiffness. Figure 6-67 shows the FEM analysis of the arms with two different materials for load of 1 N which is approximately the actual load exerted on the link. When the FullJet720 material is used which is similar to plastic the maximum displacement on the link is 300 nm. Same analysis is realized using aluminum and the maximum displacement is 10 nm.

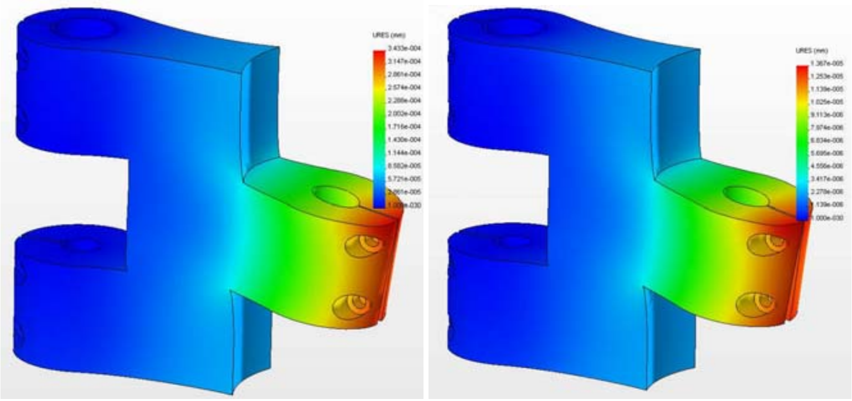


Figure 6-67 – Arm Displacement for a load of 1N (a) FullJet 720 Material (b) Aluminum 7079 Material

In addition, forward arms are thinned towards the end point of the actuator in order to decrease the weight at that point. This also provides space for the integration of the illumination system.

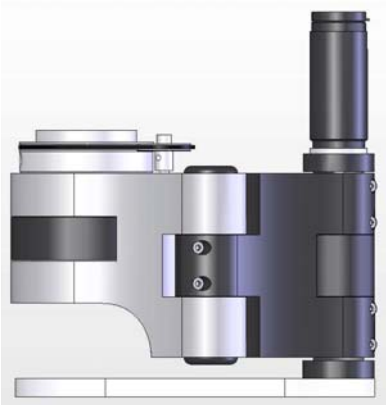


Figure 6-68 – Side View of the Design

The use of standard gears for transmission is problematic because of backlash since the stage serves for high-precision applications. A possible solution is to use anti-backlash gears which appears as a proper solution when high-precision motion is targeted. Rotational axis implemented using anti-backlash gears is shown in Figure 6-69.

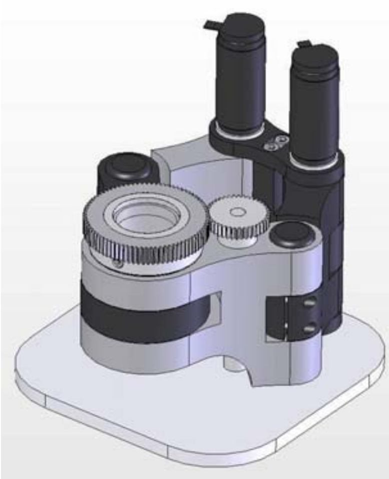


Figure 6-69 – Rotational Axis with Anti-Backlash Gears

Another solution for the implementation of the rotational axis is to use frictional belt with appropriate pulleys. Friction based actuation is good for high-precision applications since there is no backlash problem and the load (rotary platform) is not large. However, the elasticity of the belt is important and should be chosen accordingly in order to eliminate the elongation of the belt during motion. Rotational axis implemented using friction belt is shown in Figure 6-70.

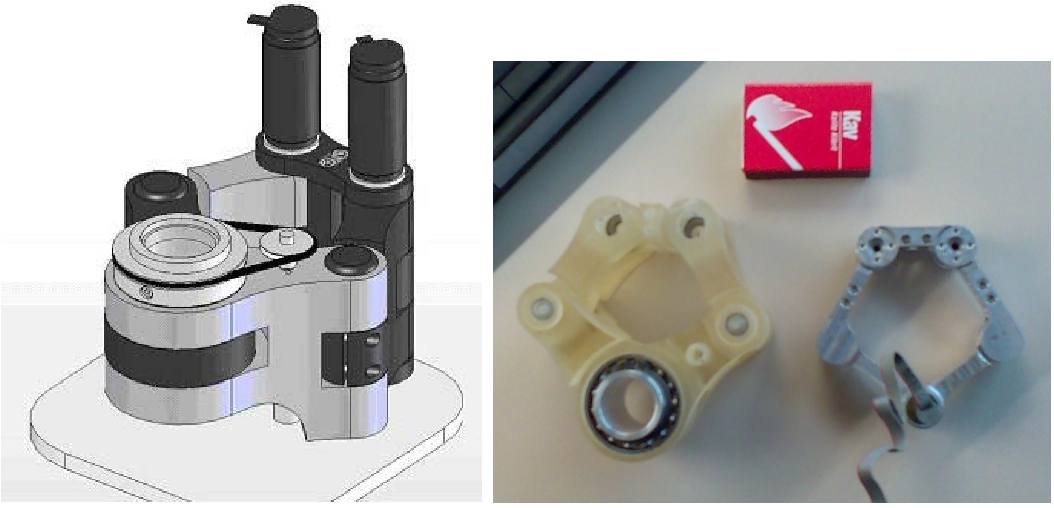


Figure 6-70 – a) Pantograph First Prototype b) First and Second Prototype

The second prototype is designed as a manipulator in order to perform pick-place operations or any other task assigned to it. At the end point with attachment of a motor, a rotational axis can be added and for the further needs a Z axis is added carrying the whole system. This prototype is produced using conventional machining which appears also as a design challenge.

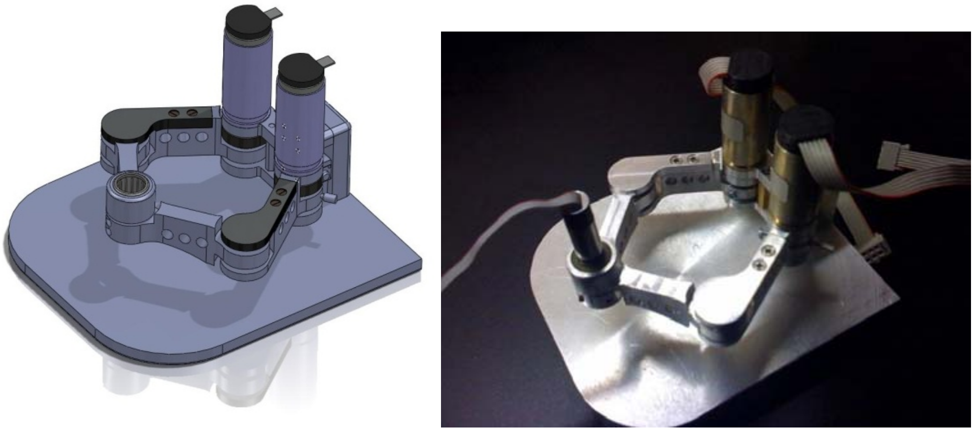


Figure 6-71 - Pantograph Second Prototype

The second prototype has some major problems related to the design and manufacturing. These problems are mainly because of the tolerances which appear at the assembly level of the mechanism. According to the design checks and some performance evaluations the second prototype is revised and some changes are made for the realization of the third prototype. These include; pre-loading of the axial bearings in order to compensate the tolerances that are especially mounted at each joint in order to allow smooth translation of motion in the presence of an axial force added to compensate for the bearing tolerances. Some improvements are also made in the design for eliminating the manufacturing tolerances and deficiencies.

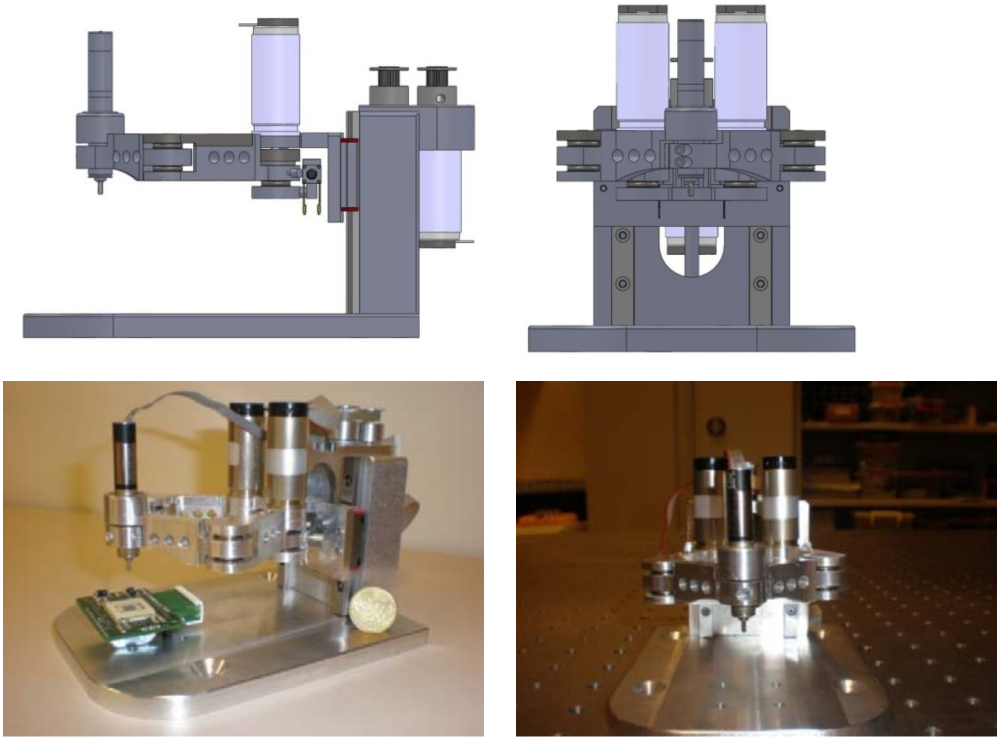


Figure 6-72 – Final Prototype

After the development of the final prototype, the mechanism is tested in order to observe the performance. The following figure shows the 10mm and 1mm circle references and the actual position of the pantograph end effector.

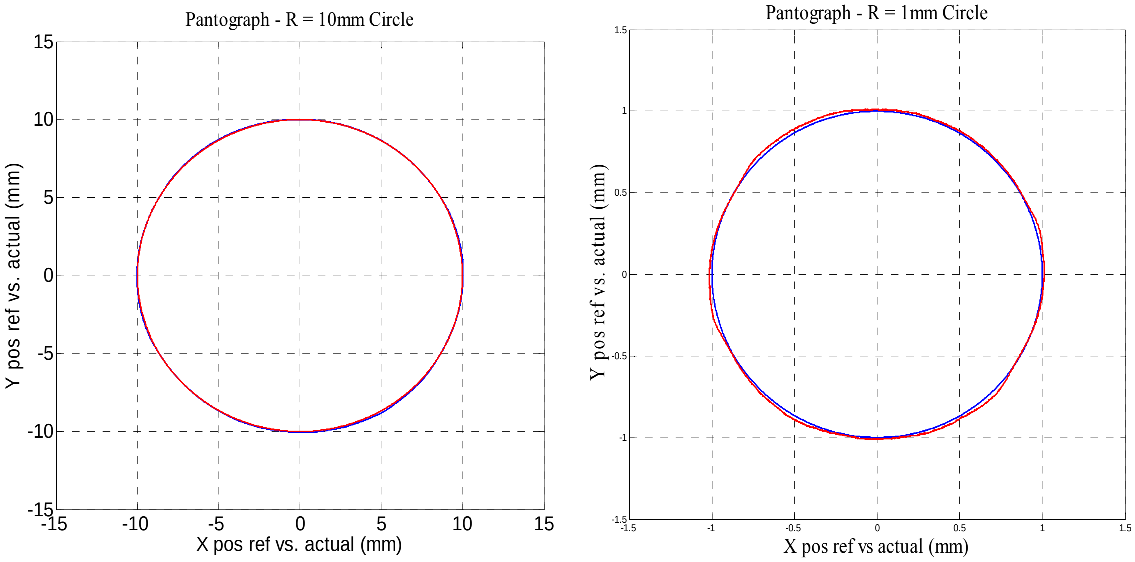


Figure 6-73 – a) 10 mm b) 1 mm radius circle reference vs. actual position

6.2.3.3 Theory and Experiments

Parallel robots that are aimed to be used for high precision applications suffer from manufacturing intolerances, assembly errors and thermal deformations. In order to decrease the sensitivity of the system to kinematical uncertainties, there is the necessity of periodic calibration with the migration of joint space measurements to task space. The unmodeled dynamics of the robot can be considered as disturbances and in order to estimate such disturbances and generate a compensating control law, a disturbance observer can be designed as depicted in [62]. The concept of migrating the measurement to task space along with disturbance estimation is realized on the pantograph in order to perform the motion control regardless of kinematical and dynamical uncertainties [63].

In order to compare the control performances through the task space and joint space measurement, an error expression is defined between the time varying reference of the end effector, $r(s)$, and the measured output, $C(s)$.

$$\xi(s) = r(s) - C(s) \tag{6-59}$$

Figure 6-74 and Figure 6-75 illustrate the block diagram representations of the control system when measurements are taken from the joint space and task space respectively.

Defining G(s), P(s) and R(s) as the transfer function of the controller and the actuators, transfer function of the linearized plant and the time varying reference in the joint space respectively, error expressions in joint and task spaces can be derived as;

$$\xi(s) = \frac{r(s)(1 + G(s) - \psi(r(s))P(s)G(s))}{1 + G(s)} \tag{6-60}$$

$$\xi(s) = r(s)\frac{1}{1 + G(s)P(s)} \tag{6-61}$$

where, ψ is a nonlinear map between the joint and task space for a given parallel robot. Assuming stability of the system, the steady state errors when measurements are taken from the joint and task space can be expressed as follows

$$\xi_{ss}(s) = \lim_{s\to 0}\frac{r(s)(1 + G(s) - \psi(r(s))P(s)G(s))}{1 + G(s)} = 1 \tag{6-62}$$

$$\xi_{ss}(s) = \lim_{s\to 0} s\frac{r(s)}{1+G(s)P(s)} \tag{6-63}$$

Taking the measurements from the task space includes the parallel robot into the closed loop of the control system as depicted in Figure 6-75. Equation (6-63) indicates that the control minimizes the steady state error in the final response by including the parallel plant inside the closed control loop. Kinematical inaccuracies including manufacturing tolerances, assembly errors, thermal deformations, etc. are included into the control loop which makes the system relatively less sensitive to these errors when compared with the joint space measurement.

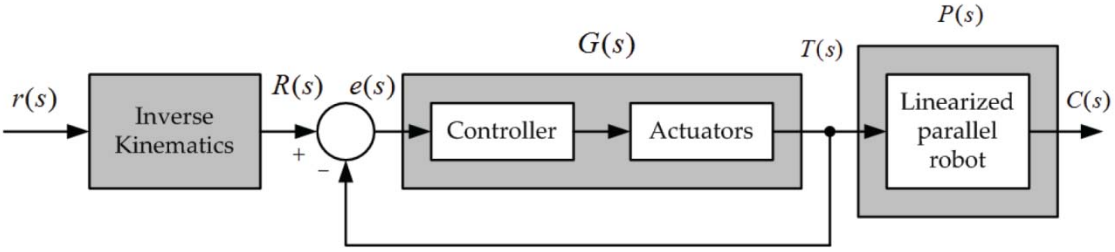


Figure 6-74 – Joint Space Measurement

The configuration level kinematics of the pantograph as depicted in Figure 6-76 can be written as follows taking the kinematical uncertainties into consideration

$$\begin{aligned} x &= (l_1+\Delta l_1)cosq_1 + (l_2+\Delta l_2)cosq_2 \\ y &= (l_1+\Delta l_1)sinq_1 + (l_2+\Delta l_2)sinq_2 \end{aligned} \tag{6-64}$$

where q_1 and q_4 are the active, q_2 and q_3 are the passive angles, x and y are the end effector coordinates of the pantograph. Δl_i is the kinematical uncertainty associated with the i^{th} link due to manufacturing tolerances, assembly errors, thermal deformations or any other factors that cannot be negligible when precise motion control is to be achieved.

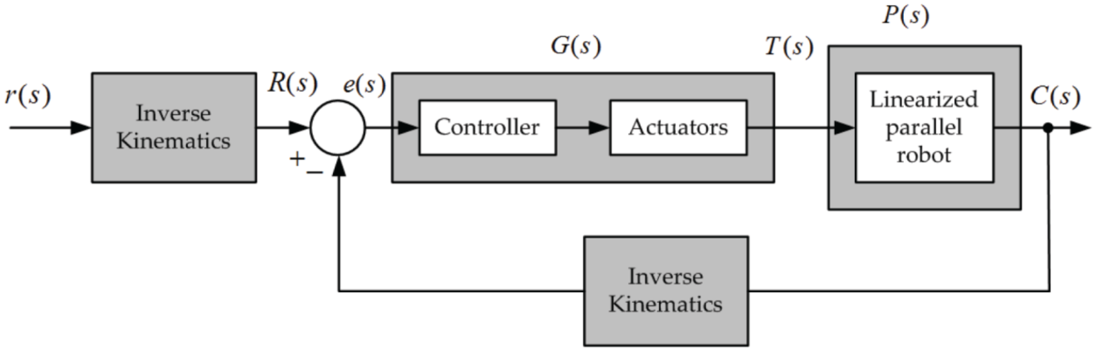


Figure 6-75 – Task Space Measurement

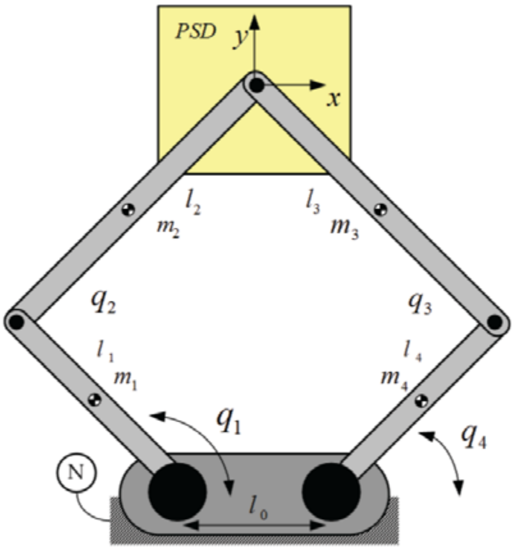


Figure 6-76 – Pantograph Configuration

The motion level kinematics can be obtained by taking the time derivative of (6-64) with respect to the Newtonian frame, N, as illustrated in Figure 6-76 and the kinematic Jacobian is derived as follows,

$$J = \begin{bmatrix} -(l_1 + \Delta l_1) sinq_1 & -(l_2 + \Delta l_2) sinq_2 \\ (l_1 + \Delta l_1) cosq_1 & (l_2 + \Delta l_2) cosq_2 \end{bmatrix} \tag{6-65}$$

The motion level kinematic relation can be expressed as

$$\underline{\dot{x}} = J\underline{\dot{q}} \tag{6-66}$$

The motion level kinematics can be integrated to obtain the configuration level inverse kinematic relation. However, integration can result in increasing the initial error with time. In that sense, feedback stabilization approach is used. Introducing the Lyapunov function as;

$$V(t) = e^T e \tag{6-67}$$

where e is the error between the right and left hand side of the configuration level kinematics equation (6-64), which can be expressed as;

$$e = \underline{x} = \Phi(\underline{q}) \tag{6-68}$$

The time derivative of the Lyapunov function

$$\dot{V}(t) = 2e\dot{e} = -2e\frac{d\Phi(\underline{q})}{dt} = -2eJ\underline{\dot{q}} \tag{6-69}$$

In order to guarantee the exponential stability, $\underline{\dot{q}}$ is selected as $J^{-1}ke$, where $k \in \mathbb{R}^{+}$ and we conclude that $\dot{V}(t) = -2ke^2 < 0$. Figure 6-77 illustrates the implementation of the inverse kinematics stabilization integration based method. The inverse kinematics is performed on the reference input along with the pantograph's end effector position measured from the task space using a position sensing device. The transformation of the task space coordinates $(\underline{x})$ to the joint space coordinates $(\underline{q})$ causes an error e_k due to the kinematical uncertainties which are given in (6-64) and (6-65). The calculation of the error $e(s)$ as given in (6-70) cancels out the error, e_k.

$$e(s) = q_{ref} + e_k - (q_{act} + e_k) \tag{6-70}$$

The implementation of the inverse kinematics stabilization integration based method makes the control system insensitive to the kinematic errors so that periodic calibration of the system becomes unnecessary.

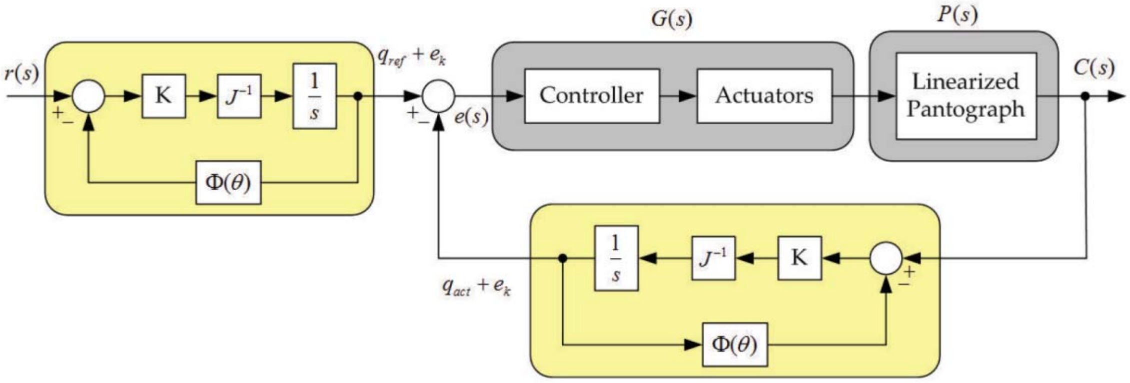


Figure 6-77 – Task Space Measurement

Recalling the equation for the dynamical equation of motion in the configuration space as in equation (6-18)

$$M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau(t) \tag{6-71}$$

Considering the dynamical inaccuracies, terms of the equation of motion can be defined as;

$$\begin{gathered} M(q) = M_n(q) + \Delta M(q) \\ b(q,\dot{q}) = b_n(q,\dot{q}) + \Delta b(q,\dot{q}) \\ g(q) = g_n(q) + \Delta g(q) \end{gathered} \tag{6-72}$$

where $M_n(q)$, $b_n(q,\dot{q})$ and $g_n(q)$ represent the nominal inertia matrix, nominal vector of coriolis and centripetal forces and the nominal gravity term respectively. The deviations from the nominal terms are represented with the Δ terms. Using a linearization feedback control law will not entirely cancel out the non-linear terms in the dynamical equation of motion. An additional control law is needed to estimate and compensate the difference between the nominal and actual plant's dynamics which can be considered as disturbance. Rewriting equation (6-71) as;

$$M_n(q)\ddot{q} + \underbrace{b_n(q,\dot{q}) + g_n(q)}_{u^{lin}(t)} + \underbrace{\Delta M(q)\ddot{q} + \Delta b(q,\dot{q}) + \Delta g(q)}_{d(t)} = \tau(t) \tag{6-73}$$

The last three inaccuracy terms can be considered as the disturbance signal $d(t)$ [64] and it can be estimated by using the low pass filter [65].

$$\hat{d}(t) = \frac{g}{s+g}[u^{ref}(t) + gM_n(q)\dot{q}] - gM(q)\dot{q} \tag{6-74}$$

where $g \in \mathbb{R}^+$ represents the observer gain controlling how fast the estimated signal converges to the actual disturbance [66]. Estimated disturbance is used to generate the control law given in (6-75) and used with the feedback linearization control law, $u^{lin}(t)$, to achieve the overall control law depicted in Equation (6-76).

$$u^{dist}(t) = \frac{1}{k_{tn}}\hat{d}(t) \tag{6-75}$$

$$\tau(t) = u^{lin}(t) + u^{dist}(t) + u^{ref}(t) \tag{6-76}$$

The block diagram implementation of the overall control law is depicted in Figure 6-78. The first term in (6-76) is the feedback linearization control law depending on the nominal plant dynamics, the second is generated from the disturbance observer and used to cancel out the nonlinear terms in the pantograph's dynamic equation of motion and the third one is an arbitrary control law.

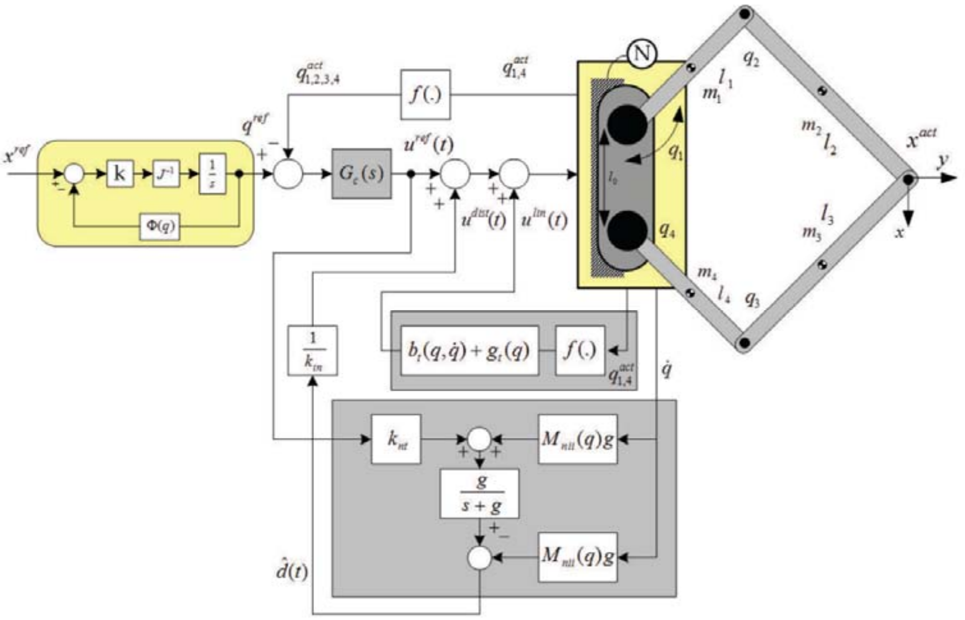


Figure 6-78 – Configuration Space Measurement

The dynamical equation of motion in the task space is written as follows

$$M_t(q)\ddot{x} + \ b_t(q,\dot{q}) \ + \ g_t(q) \ = \ F_t(t) \tag{6-77}$$

that can be obtained through the following mappings [67]

$$\begin{aligned} M_t(q) \ &= \ (J^\dagger)^T M(q) J^\dagger \\ b_t(q,\dot{q}) \ &= \ (J^\dagger)^T b(q,\dot{q}) \ - M_t(q) J\dot{q} \\ g_t(q) \ &= \ (J^\dagger)^T g(q) \\ \tau(t) \ &= \ J^T F_t \end{aligned} \tag{6-78}$$

where $F_t(t)$ and $J^\dagger$ are the task space force vector and the Jacobian matrix pseudo-inverse, respectively. Similar to the kinematical uncertainties argument, (6-77) can be rewritten taking the following dynamical uncertainties into consideration

$$M_t(q)\ddot{x} + \ b_{tn}(q,\dot{q}) + \Delta b_t(q,\dot{q}) + \ g_{tn}(q) + \Delta g_t(q) \ = \ F_t(t) \tag{6-79}$$

Similar with (6-74), the linearization feedback control law is defined as

$$u_t^{lin}(t) = b_{tn}(q,\dot{q}) + g_{tn}(q) \tag{6-80}$$

The disturbance observer structure is modified considering the measurements are taken from the task space

$$\widehat{d_t}(t) = \frac{g}{s+g}\left[u^{ref}(t) + gM_{tn}(q)J\dot{x}\right] - gM_{tn}(q)J\dot{x} \tag{6-81}$$

Associated with the control law

$$u_t^{dist}(t) = \frac{1}{k_{tn}}\widehat{d_t}(t) \tag{6-82}$$

And the overall control law is written as

$$F_t(t) = u_t^{lin}(t) + u_t^{dist}(t) + u_t^{ref}(t) \tag{6-83}$$

The block diagram implementation of the control law for the system where the measurement is realized in the task space using a position sensing device is illustrated in Figure 6-79. Since the device directly measures the end effector's actual position, the system becomes free from the kinematical or dynamical inaccuracies.

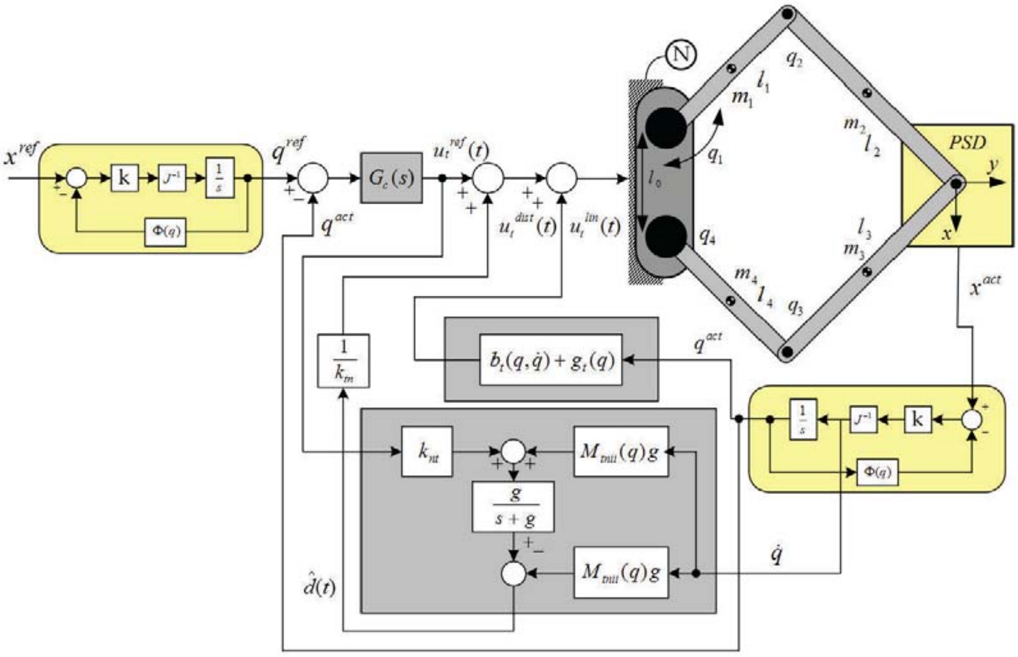


Figure 6-79 – Task Space Measurements

The concept is tested by using the pantograph as the parallel manipulator. For the joint space measurement, 512 pulse/rev resolution optical encoders at the actuated joints are used and a xy position sensing device located at the end effector measuring the position of the laser beam attached to the end effector of the pantograph is used. The experimental setup is shown in Figure 6-80.

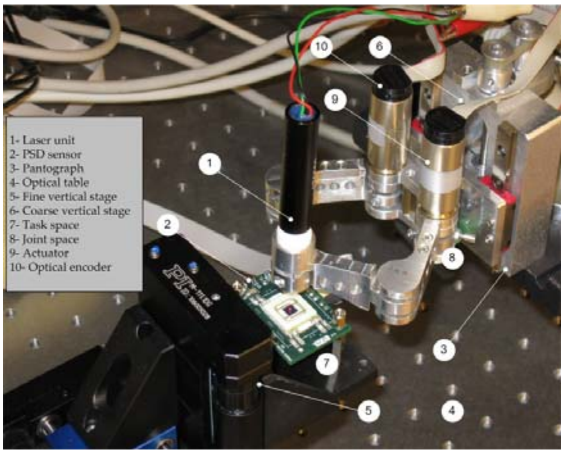


Figure 6-80 – Experimental Setup

Figure 6-81 and Figure 6-82 show the result of the experiments for a circular reference trajectory with 100 μm diameter for the joint and task space measurements respectively.

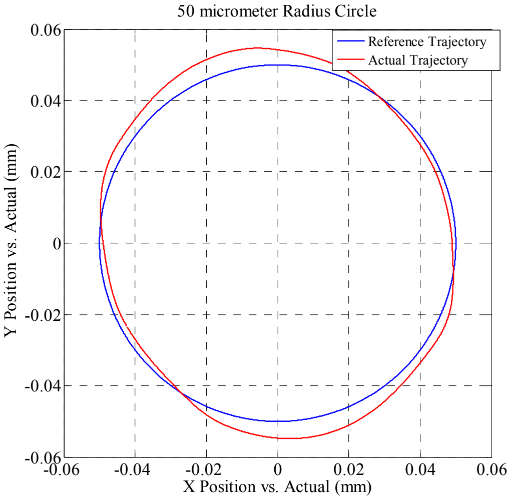


Figure 6-81 – 100 micrometer circle reference and actual trajectory (configuration space measurement)

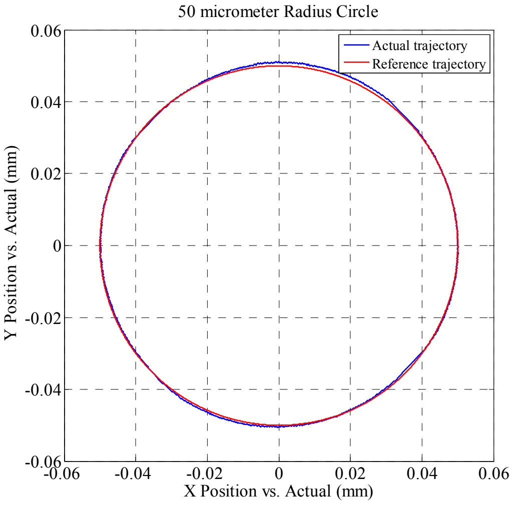


Figure 6-82 – 100 micrometer circle reference and actual trajectory (task space measurement)

There is a significant error in the joint space measurement system since the entire parallel robot is placed outside the control loop. The control law only guarantees that the robot's active angles follow reference trajectories but has no effect on the steady state error as a result of the kinematical inaccuracies. As it can be seen in Figure 6-82, the pantograph follows the circular reference trajectory regardless of the kinematical inaccuracies. Kinematical inaccuracies are included inside the closed loop rather than keeping them outside when measurements are taken from the active joints.

6.2.4 Micromanipulator

The micromanipulator developed for the microassembly workstation is designed in such a modular way that it can be used in the further developed systems. As a part of the microassembly module, the micromanipulators allow the positioning of the end effector with very high accuracy. The manipulators mainly consist of three translational degrees of freedom in x, y and z coordinates which makes it capable of executing simple pick and place tasks and some 2D tasks such as pushing, pulling type of assembly tasks. The main design consideration for the manipulator unit is that it should have enough travel distance to cover the necessary workspace area while having enough accuracy smaller than 10 micrometers. Travel range, accuracy and the speed parameters are considered for the selection of the stages that are used for the micromanipulator. For the modular and compactness of the micromanipulator design the size of the positioning stages is another issue to be considered.

The micromanipulator is enhanced with a versatile tool holder which is realized for the efficiency and flexibility of the assembly tasks. The tool holder mounted on the micromanipulator is shown in Figure 6-83 (b). It is designed in such a way that the holder can be moved keeping the tool tip as the pivot point and be fixed with any angle. It can be rotated also for the easy removal of the manipulation tools.

The microassembly tasks may require better accuracies than the micromanipulator can provide. In order to provide better accuracies, a high precision stage is integrated to the micromanipulator which formed a structure of XYZ coarse-fine positioning system. With that configuration, coarse positioning stage provides large workspace coverage while providing enough positional accuracy to bring it into the range of motion of the fine positioning stages and the final positioning is made by the fine positioning stage with high accuracy. Submicrometer resolution, workspace of hundreds of micrometers, compact size and high speed of response are the important parameters to be considered for the selection of the high precision stage. Piezo actuators are a suitable choice since they offer high precision in the range of a few nanometers and they can also handle high loads and their fast response. On the other hand, they have limited travel ranges and suffer from hysteresis and drift since the former one can cause not only positioning error but also instability.

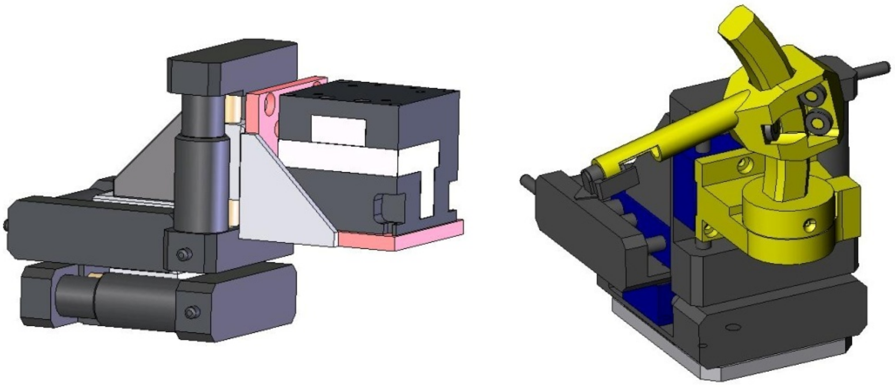


Figure 6-83 – Manipulator Configuration (a) Coarse-Fine (b) Coarse

In addition to the manipulation stages with the configuration of coarse and fine positioning with orthogonal axes of x, y and z, the system also has a 3 DOF sample precision positioning system (x,y,θ) which provides the usage of the substrate into the field of view of the microscope. Rotational stage is designed over a xy Cartesian positioning system with the resolution of 45 nano degrees. The sample stage is also designed in such a way that it allows backlighting with a gap opening of 20 mm. The sample positioning system is shown in Figure 6-84.

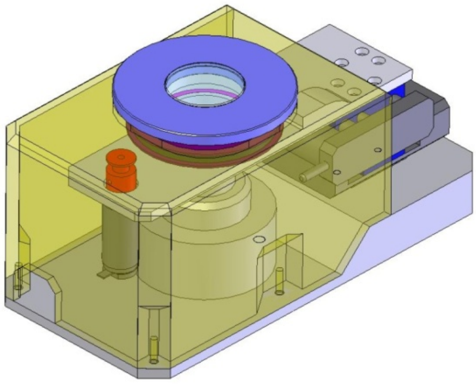


Figure 6-84 – Sample Positioning System

6.2.4.1 Motion Control

For the systems where high precision motion control is required, robustness of the control algorithm is the most crucial factor. Moreover, if the plant has high nonlinearities such as hysteresis in piezo actuators or friction, using a robust controller designed according to the nominal plant parameters and which rejects parameter uncertainties would be advantageous. Variable structure control with sliding modes, frequently named as sliding mode control, is characterized by a discontinuous control action which changes structure upon reaching a set of predetermined switching surfaces. This control may result in a very robust system with its built-in disturbance rejection, which in turn implicitly compensates for the unmodeled dynamics, and thus provides a possibility for achieving high precision and fast response.

Discrete time implementation of sliding mode control for the system is as follows;

Considering the system:

$$\dot{x} = F(x,t) + B(x,t)u \tag{6-84}$$

where $x \in \mathbb{R}^n$ is the state vector of the system. $F(x,t)\colon \mathbb{R}^n x \mathbb{R}^+ \rightarrow \mathbb{R}^n$ is a continuous and bounded linear or nonlinear function which defines the uncontrolled dynamics of the system. $B(x,t)\colon \mathbb{R}^n x \mathbb{R}^+ \rightarrow \mathbb{R}^{nxm}$ is a continuous and bounded matrix with $rank(B) = m$ for every x, t couple, yielding the system to be linear according to the control input. $t \in \mathbb{R}^+$ is the independent time variable. Derivation of the control law starts with the selection of the positive definite Lyapunov function candidate, (σ) satisfying the Lyapunov stability criterion;

$$\dot{v}(\sigma)v(\sigma) < 0 \tag{6-85}$$

For a Lyapunov function of the form

$$v(\sigma) = (\sigma^T\sigma)/2 \tag{6-86}$$

Derivative of the function

$$\dot{v}(\sigma) = \sigma^T\dot{\sigma} \tag{6-87}$$

Designing the control function such that

$$\dot{\sigma} + D\sigma = 0 \tag{6-88}$$

Derivative of Lyapunov function becomes a negative definite function as

$$\dot{v}(\sigma) = -\sigma^T D\sigma \tag{6-89}$$

Satisfying the Lyapunov stability criterion, where $D \in \mathbb{R}^{mxm}$ is a positive definite symmetric matrix defining the slope of the sliding manifold at each dimension. Lyapunov function and its derivative having opposite signs with the aid of control enforce the system to move to $\sigma = \dot{v}(\sigma) = 0$ ensuring stability.

For the discrete time sliding mode developments continuous motion equation should be replaced by its discrete time equivalent

$$x_{k+1} = F_k(x_k) + B_k(x_k)u_k \tag{6-90}$$

for $x_i = x(i\Delta t)$ $x \in \mathbb{R}^n$, $F_i = \Delta t F(x_i, i\Delta t) + x_i$ $F \in \mathbb{R}^n \rightarrow \mathbb{R}^{nxm}$, $B_i = \Delta t B(x_i, i\Delta t)$ $B \in \mathbb{R}^n \rightarrow \mathbb{R}^{nxm}$, $u_i = u_i(\Delta t)$ $u_i \in \mathbb{R}^m$, $i \in Z^+$and Δt is the sampling time.
For a tracking error $e_x = x_{ref} - x$, σ is selected as $\sigma(x) = Ge_x$ for $G \in \mathbb{R}^{mxn}$ such that $det(GB_k) \neq 0$ to satisfy control objectives on the sliding manifold $\sigma(x) = 0$.

$$\dot{\sigma} = GB(u - u_{eq}) \tag{6-91}$$

solving for u_{eq}

$$u_{eq} = u - [GB]^{-1}\dot{\sigma} \tag{6-92}$$

Since u_{eq} is a continuous function, approximation of the current value of u_{eq} yields to

$$u_{eqk} \approx u_{eqk-1} = u_{k-1} + [GB_k]^{-1}\dot{\sigma}_{k-1} \tag{6-93}$$

Writing $\dot{\sigma}$ in discrete form using Euler's approximation

$$\dot{\sigma}_{k-1} = \sigma_k - \sigma_{k-1}/\Delta t \tag{6-94}$$

putting (6-91) and (6-88) and solving for u_k

$$u_k = u_{eqk} - [GB_k]^{-1} D\sigma_k \tag{6-95}$$

using the approximation

$$u_k = u_{eqk-1} - [GB_k]^{-1} D\sigma_k \tag{6-96}$$

solving (6-93) and (6-94) together

$$u_{eqk} \cong u_{k-1} - [GB_k]^{-1}(\sigma_k - \sigma_{k-1})/\Delta t \tag{6-97}$$

putting (6-97) into (6-96)

$$u_k = u_{k-1} - [GB_k]^{-1}((D + 1/\Delta t)\sigma_k - \sigma_{k-1}/\Delta t) \tag{6-98}$$

simplifications yield to

$$u_k = u_{k-1} - [GB_k\Delta t]^{-1}((D + 1/\Delta t)\sigma_k - \sigma_{k-1}) \tag{6-99}$$

The control structure (6-99) is suitable for implementation, since it requires measurement of the sliding mode function and the value of the control applied in the preceding step. Thus, (6-99) is used as control structure as discrete sliding mode for translational stages and piezo actuation.

The control method is tested on micro translational stages and the step response of the stages is shown in the Figure 6-85 for 1 micron.

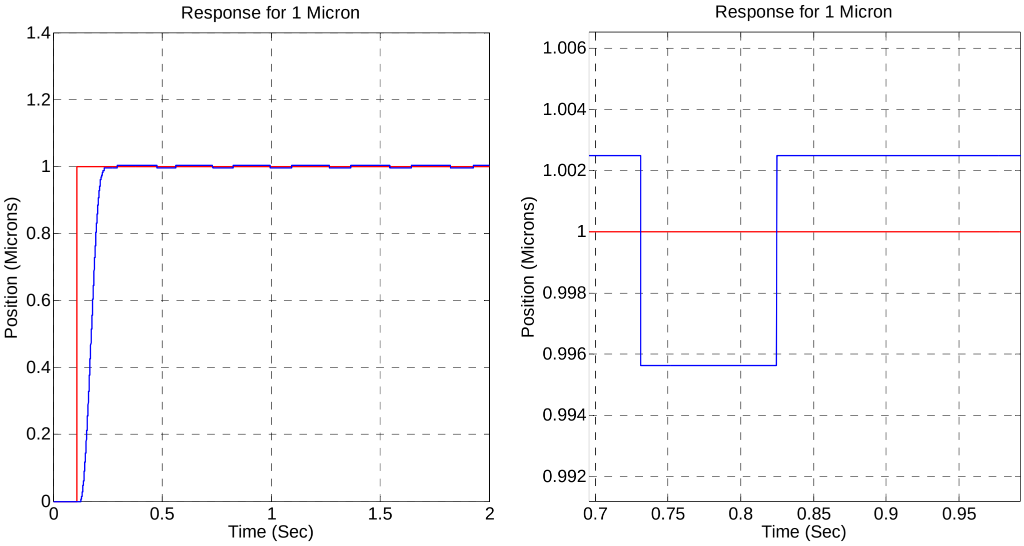


Figure 6-85 - Step response of translational stages for 1 µm

As it can be seen from the figures that for the given position references, performance of the translational stages is satisfying since the responses are good but only suffering from a 0.007 microns oscillation representing the resolution of the encoder. That amount of error can be neglected in our case since it can be compensated by the fine translational stages if necessary.

Disturbance observer implementation (Figure 6-86) for the piezo actuated stages is explained in [68] and directly used in the system.

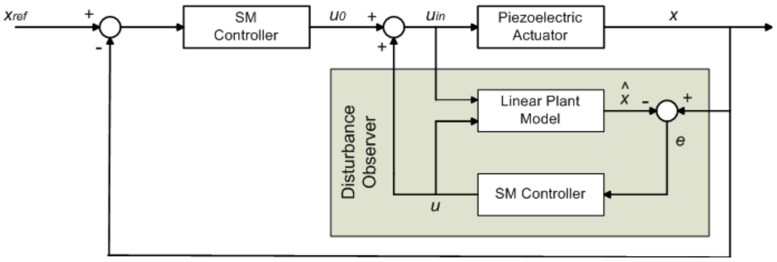


Figure 6-86 – Observer Implementation

The resulting step response of the piezo stages for 10 nm reference is shown in Figure 6-87. As seen in the figure, system is able to achieve the desired position with a fast rise time. However the system suffers from the noises belonging to high frequency range resulting from the measurement devices which affect the steady state of the system and forces an oscillatory behavior with maximum amplitude of 1-1.5 nanometer.

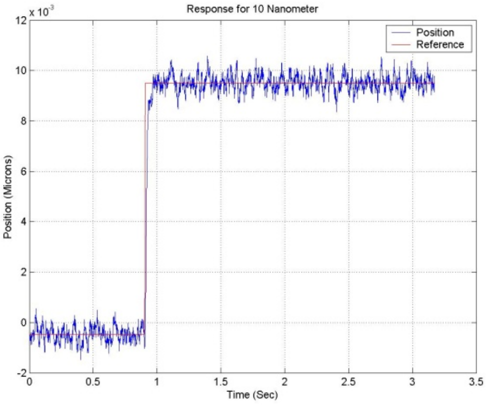


Figure 6-87 – Step response of piezo stages for 10nm

Determination of the joint inputs required causing the end effector to execute desired motions which are generally specified as a sequence of positions or as a continuous path is the main control problem for robotic manipulators. These joint inputs can be changed according to the model used for the controller design. They can be joint forces and torques or voltage inputs to the actuators. The control method used can significantly influence the performance of the manipulator according to the type of the application in which the manipulator will be used. The mechanical design of the manipulator (kinematic configuration) will also affect the type of the control scheme needed.

6.2.5 System Conveyor

A carriage unit is necessary for the assembly system since the parts should be carried from one station to the other for different processes to take place in the assembly module. The carriage unit specifications are limited with the size of the module and the working space of the manipulators which will be used. According to the dimensions an 800 mm conveyor with 60 mm width and 30 mm height is integrated to the system as the system conveyor. The conveyor has maximum speed of 9m/min and a belt width of 50 mm on which it carries the trays designed for carrying the parts in between the stations for inspection and assembly. The conveyor is equipped with stoppers for the positioning of the trays at the workspace of the stations and with sensors detecting the presence of the trays at each station. The speed of the conveyor can be adjusted automatically from the user interface since it is driven using a developed dc motor driver. The system carriage unit, trays, stoppers and sensors are shown in the following pictures.

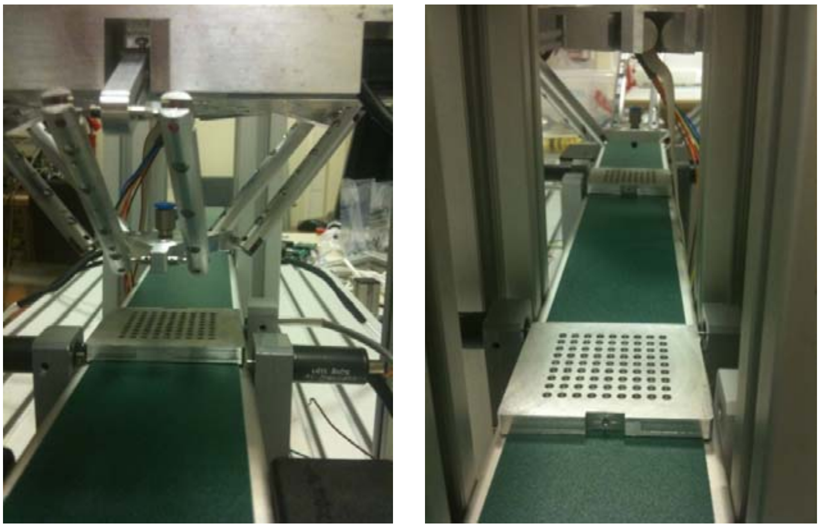


Figure 6-88 – System Carriage Unit

Stoppers that are used in the system are 15V actuated solenoids which are located on the carriage unit to position the trays and block their motion when actuated. Sensors are 24V output inductive sensors detecting the presence of the trays at the process stations to give the necessary information for the system flow.

6.2.6 Vision Unit – Microscope

A vision sensor is necessary for the assembly module in order to visualize the parts to be assembled and for the system to acquire the position and orientation of the parts. The vision system is used to give necessary feedback to the main system supervision unit in order to realize the assembly process. Since the robotic assembly module is a small standalone unit, the size of the vision sensor that would be used in the system is one important aspect. Sizes of the parts are small in the micrometer to millimeter range so that in order to detect the parts precisely magnification is also important. As a result of that, a microscope is needed as the vision sensor in the system. TIMM 400, a miniaturized video microscope with its small size (155 mm x 22 mm) and the low weight (approx. 100g) is selected to be used as the vision sensor for the module. With its continuous variation of magnifications between 0.1 and 400 times without an objective changing and with the attachment of various additional optical components the magnification can be increased until to the limit of the light diffraction. Integrated CCD-Camera enables the capturing of the microscope image and the image can be used for image processing purposes in order to detect the parts to be assembled. A ring light is attached to the microscope for the illumination of the workspace for a clear and homogeneous image for precise detection of the parts using image processing techniques. The vision station is a closed black box unit to maintain the homogeneity of the illumination. The vision station and the ring light attached to the microscope are shown in the following pictures.

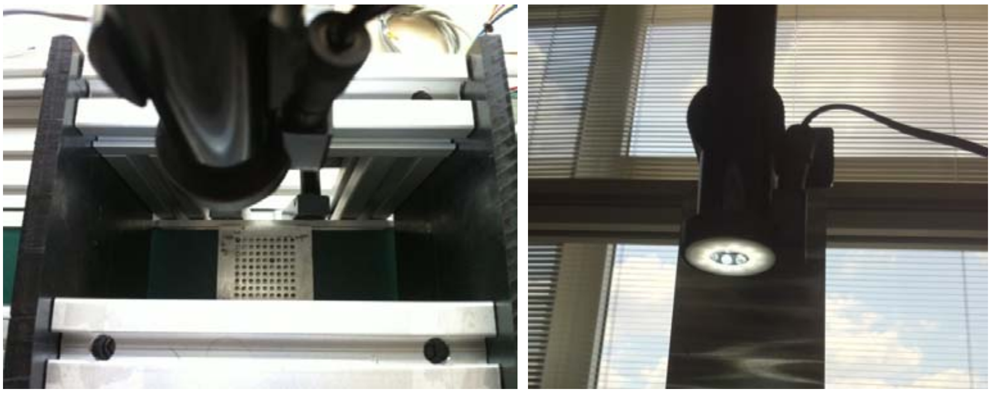


Figure 6-89 – (a) Vision Station (b) Ring Light

6.3 Experiments

In order to test the reliability and performance of the system, pick-place experiments using 3mm diameter steel balls are realized in two different configuration of the microfactory assembly module. The first configuration with the FPGA and the other with dSpace PPC board as the control system are defined in the previous sections.

For the experiments, a grid structured tray is designed to carry the steel balls in between the stations of the assembly module (Figure 6-90). The experiments are realized using a vision sensor to detect the positions of the steel balls, a conveyor as the carriage unit and a Delta robot for the realization of the pick place tasks as the inner modules of the assembly process module. The conveyor is equipped with sensors and stoppers for the positioning of the trays at the stations.

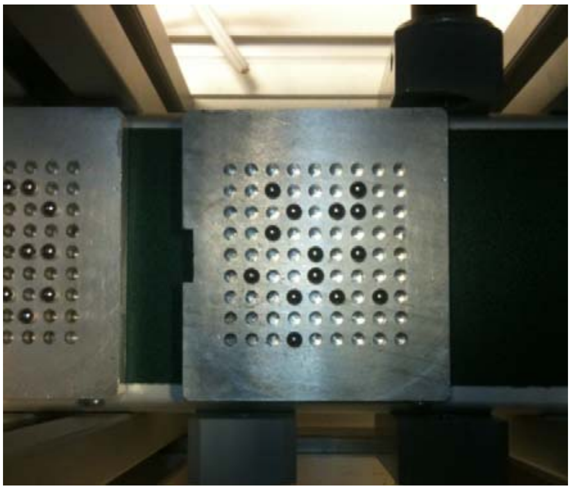


Figure 6-90 – Grid Structured Tray

The overall system operations are controlled using the GUI and the operating procedure of the µfactory is defined as follows for the pick place experiments;

- Feed loaded tray on the conveyor (The command is given to the system by a Conveyor Loaded/Palette Fed Button on GUI)
- Conveyor Loaded Command → Conveyor ON (Conveyor starts working) and Vision Station Stoppers ON (Stoppers at the vision station are activated in order to hold the palette in position at the station)
- Each station is equipped with solenoids in order to stop/keep the tray in position at the station and inductive sensors to check if the tray exists or not. If vision palette

sensor ON → Conveyor OFF (conveyor stopped for a stable image for image processing)

- Process Image (The tray image is processed in order to locate the steel balls for position determination and task generation)
- The positions of the existing parts are displayed on the grid next to the image in GUI (Figure 5-9). Detected parts are marked with the color green on the grid structure. Now the system is ready for task generation by the user. The user realizes the task generation by simply clicking on the existing part and the desired destination point on the grid structure. For each task defined, source and destination points are displayed on the grid structure and the position of the grids are displayed numerically just below the grid. The user adds each subtask to the task list using the related command button and each desired subtask is listed and shown on the GUI. When the task generation is completed, the "Execute Task" command is given to the system using the GUI.
- Execute Task → Vision Palette Stoppers OFF (in order to release the tray from the vision station on the conveyor) → Assembly Station Stoppers ON (in order to keep the tray at the assembly station) → Conveyor ON
- If the assembly station sensor ON (tray is located at the assembly station) → Conveyor OFF
- According to the task list that is generated by the user, the Delta robot starts to generate each subtask one after another with the procedure defined below until all the subtasks are realized
    - GOTO HOME + Zoffset position
    - Vacuum ON
    - GOTO HOME position
    - GOTO HOME + Zoffset position
    - GOTO DEST + Zoffset position
    - GOTO DEST position
    - Vacuum OFF
    - GOTO DEST + Zoffset position
- Assembly DONE → Assembly Station Solenoids ON → Conveyor ON (until the tray reaches the final position)

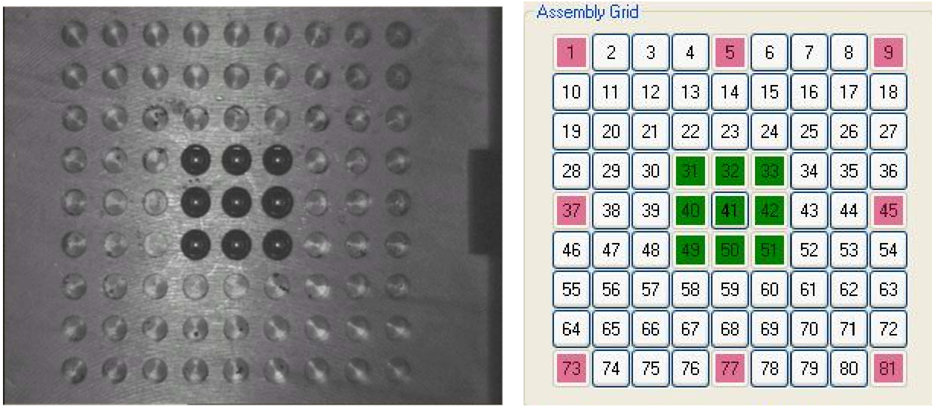
The GUI is designed in order to control the whole experimental procedure which is explained above. In addition to those listed, it also provides;

- Monitoring some of the parameters like the position of the Delta robot,

- Sending independent commands for each unit of the microfactory like; Solenoids ON/OFF, Conveyor ON/OFF, Delta Robot Position Reference, etc.
- Providing the visual feedback acquired from the vision sensor

Each unit in the factory can be controlled and the outputs can be monitored from the GUI since it is used for the development phase of the microfactory.

A simple 8 step pick place task defined using the GUI is shown in Figure 6-91. The image captured from the vision sensor is shown in Figure 6-91(a). As it can be seen in the image, nine steel balls are located in the center of the grid structured tray. Figure 6-91(b) shows the GUI interface for the task generation. The green grids show the locations of the detected steel balls and the pink grids show the destination points defined by the operator. So the whole task is composed of 8 pick place subtasks to be generated by the Delta robot.



Assembly Grid

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

Figure 6-91 – (a) Tray Image (b) GUI Grid

The following pictures show the Delta robot in action during the realization of the task.

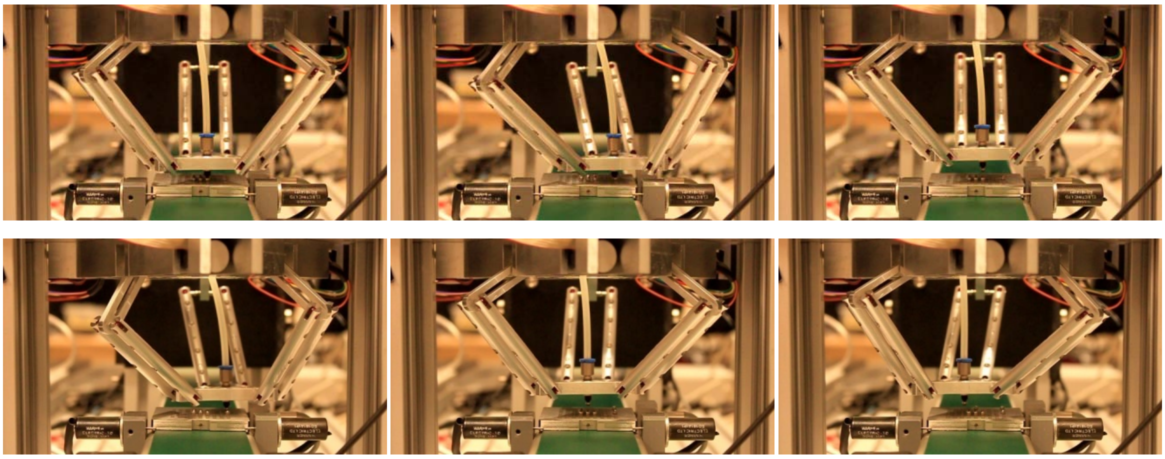


Figure 6-92 – Delta Robot Generating the Pick Place Tasks

Pick Place experiments are first realized on the assembly module with the FPGA control hardware. The following figures are showing the FPGA control system performance for the 8 steps pick place task experiments defined above and shown in Figure 6-91. The experiments are realized first keeping the speed of the Delta robot low and then the speed is increased and the same task is repeated. The first experiment is realized with 10% of maximum speed limit and the motion graphs of the Delta robot are shown in Figure 6-93. In the figure, the position of each arm of the robot, X and Y positions separately and each position of the home and destination points on the XY motion plane which is a projection of the grid structure are shown. As it can be seen in the XY motion graph the robot follows a linear trajectory in between the home and destination points.

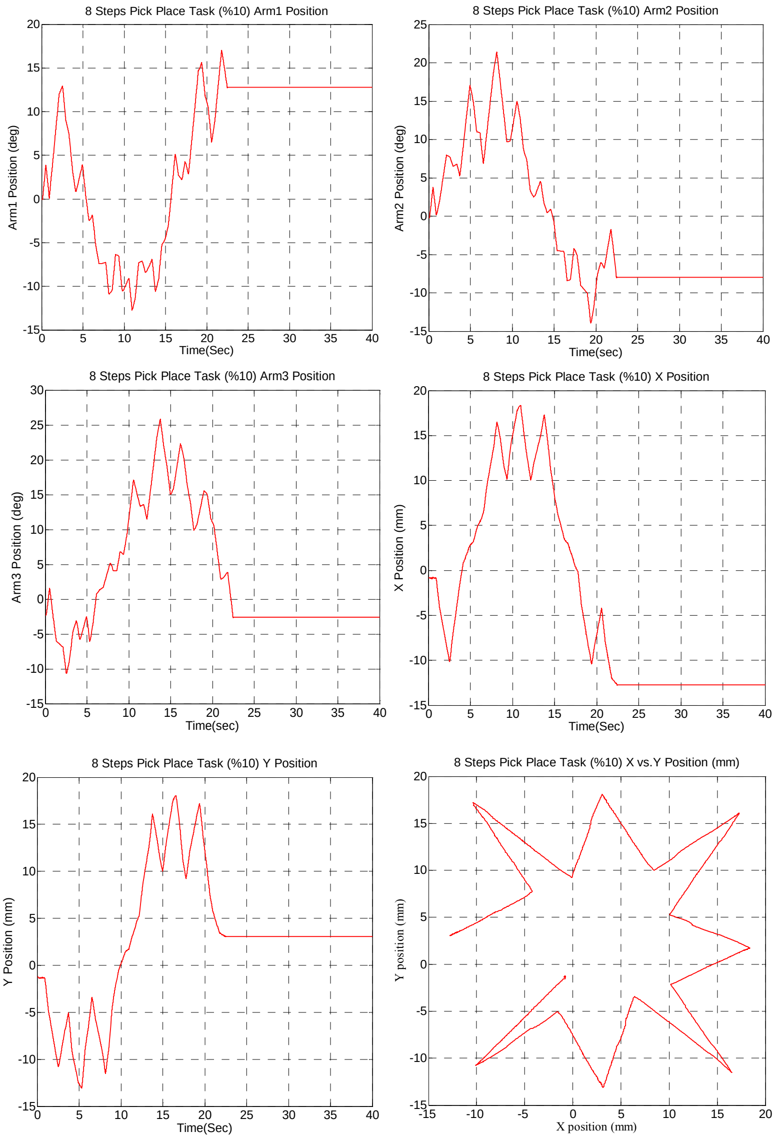


Figure 6-93 – 8 Steps Pick Place Experiment (10% Speed with FPGA)

Figure 6-94 shows the same 8 Steps Pick Place task with %50 speed performance. In that case, the lines are not smooth/linear as it was in the experiments with %10 speed since the increase in speed decreases the control performance decreases. Also in FPGA the trajectory generation procedure is weak as a result of the performance of the processor of the FPGA which resulted in such a motion. At pick place locations (home and destination points), parasitic behaviors occurred as a result of the control performance that can be achieved with FPGA.

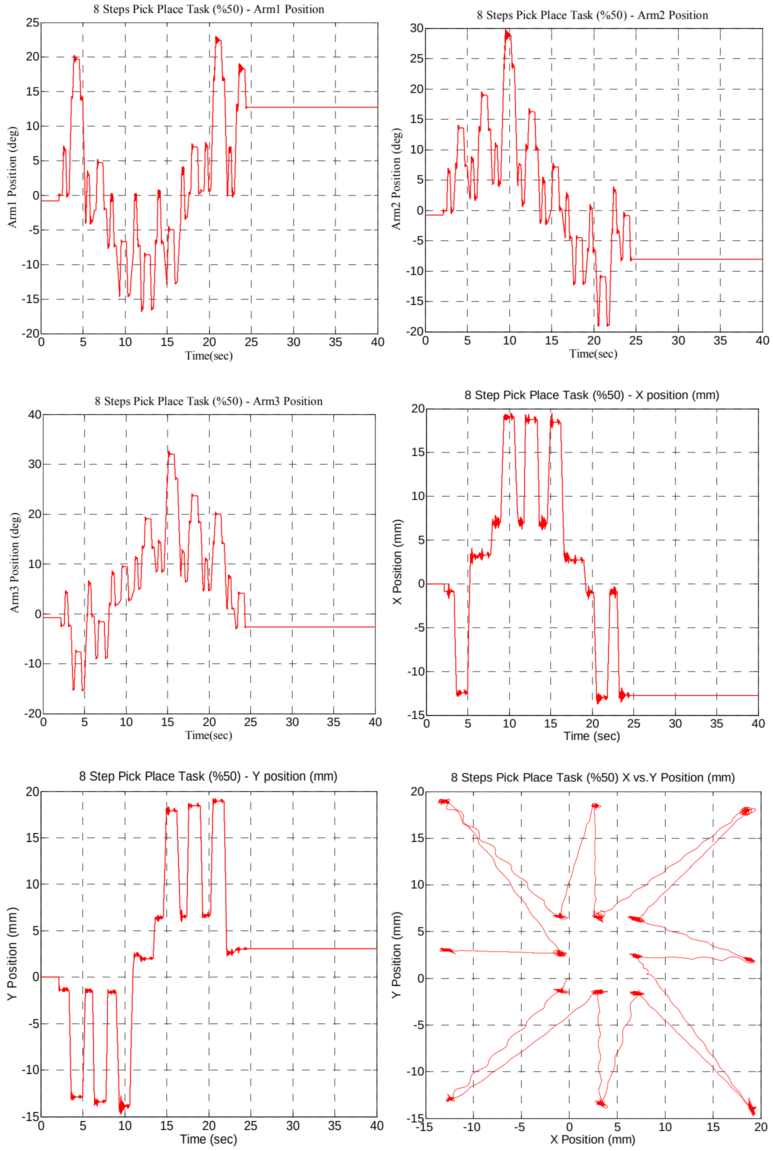


Figure 6-94 - 8 Steps Pick Place Experiment (50% Speed with FPGA)

The final experiment is realized at the maximum speed and the results are shown in Figure 6-95. The motion again suffers from the trajectory generation weakness and the control performance. The bottleneck for the cycle time of the task is the idle times at the pick and place positions as a result of the vacuum generation. It can be clearly seen in X or Y position graphs in between the motions. For the accurate placement of the parts, an idle time without motion is added for the vacuum generation and extermination in order to place the parts accurately.

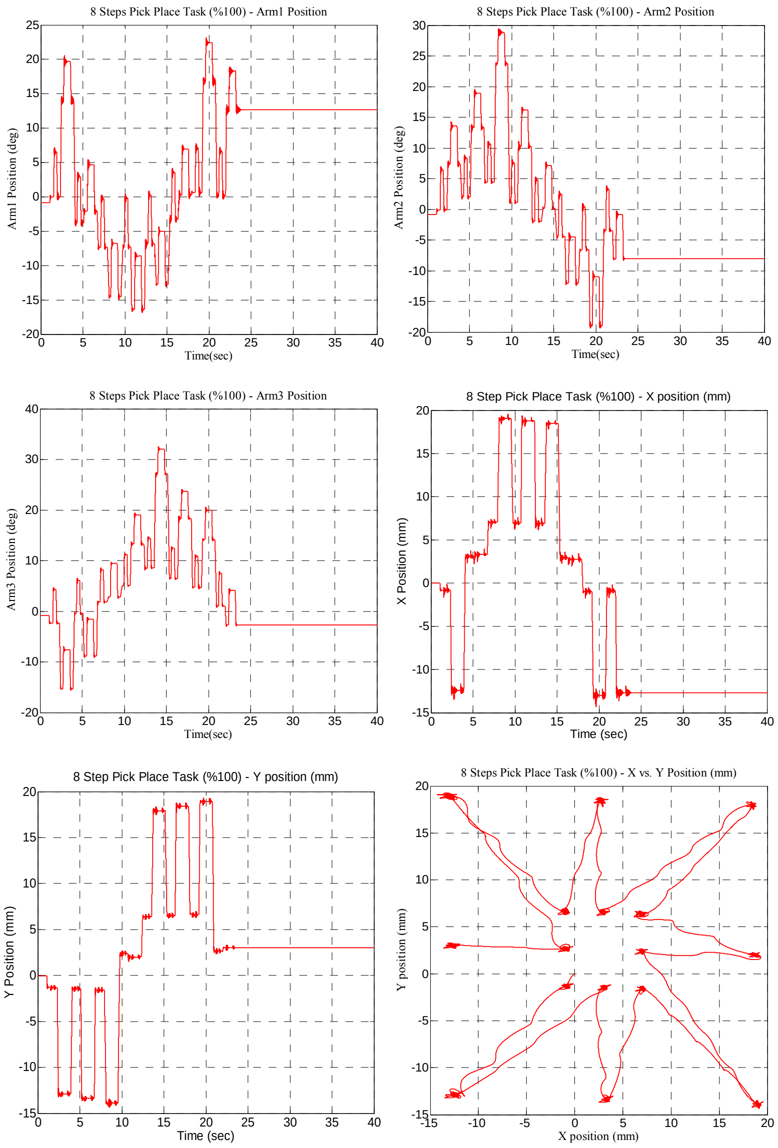


Figure 6-95 - 8 Steps Pick Place Experiment (100% Speed with FPGA)

The performance of the experiments realized with FPGA oriented us towards configuring another control unit for the system. The system is reconfigured using ds1103 PPC board as the control system and the same pick place experiments are realized in order to see the difference in the performance. As it can be seen from the figures below, the control system allows the implementation of the trajectory generation algorithm so that the Delta robot can perform a linear smooth motion between the home and destination points regardless of the increase in speed. Since the speed is increased in each experiment the control parameters are adjusted accordingly in order to get rid of the overshoots at the points. Figure 6-96 shows the motion of the Delta robot for the experiment realized at 60mm/sec speed. The motions of the arms, X, Y and Z motion separately and the XY plane motion are shown in each figure set.

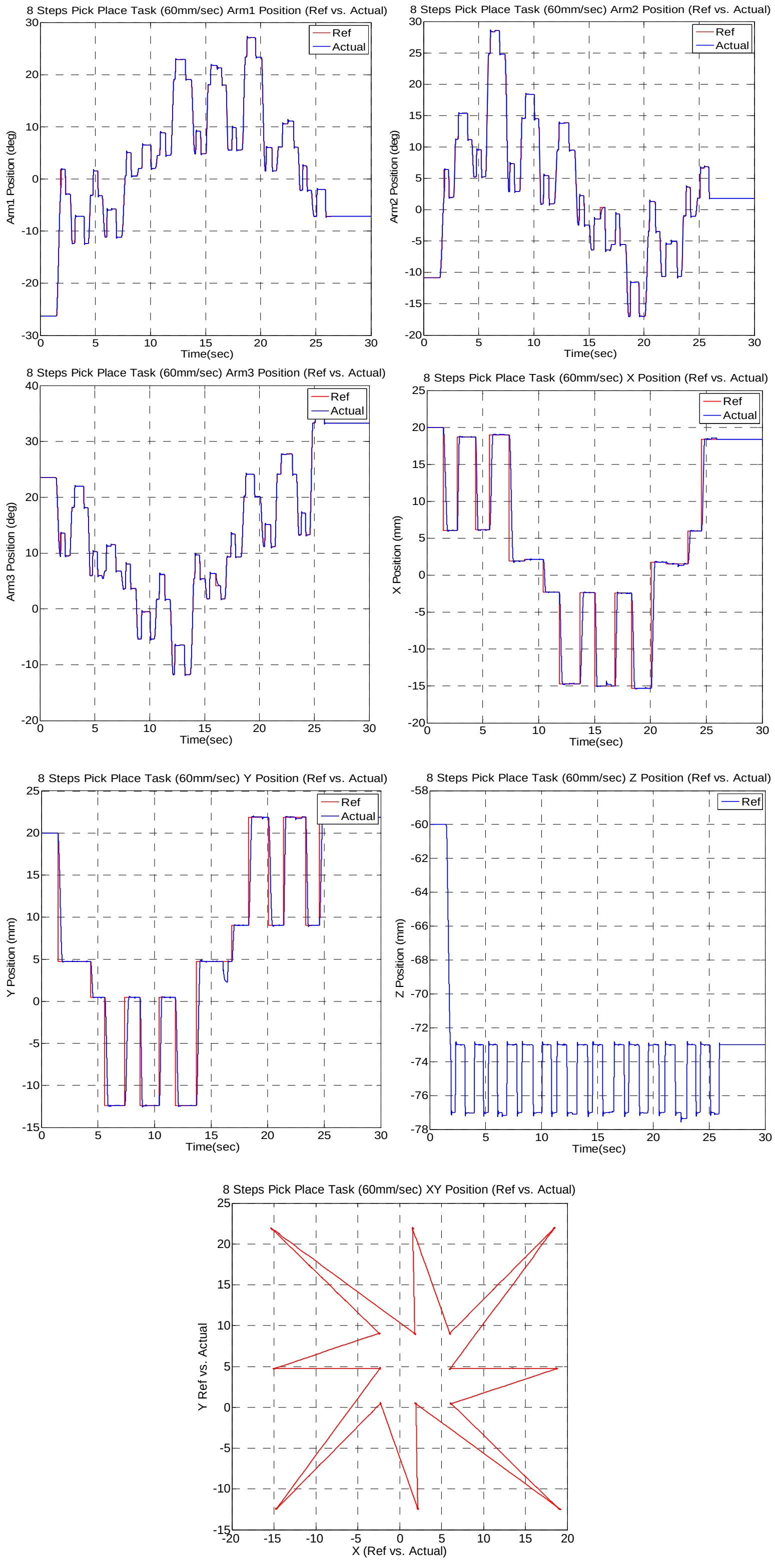


Figure 6-96 - 8 Steps Pick Place Experiment (60mm/sec Speed with ds1103 Controller Board)

As it can be seen in the figures, reference trajectories can be followed precisely even with the increasing speed. Figure 6-97 shows the motion at a speed of 120mm/sec and resulting trajectory is linear and smooth which was not the case with the FPGA experiments.

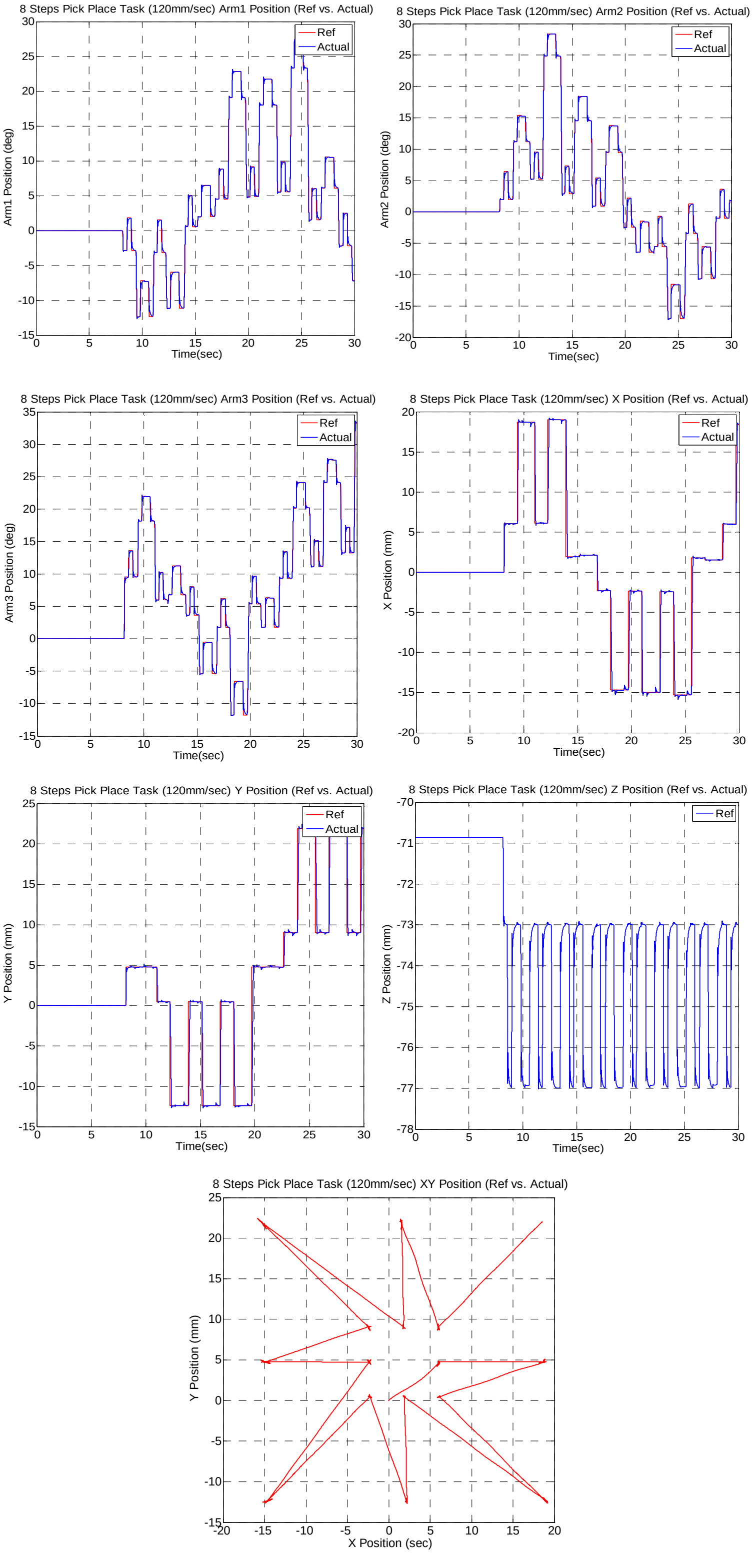


Figure 6-97 - 8 Steps Pick Place Experiment (120mm/sec Speed with ds1103 Controller Board)

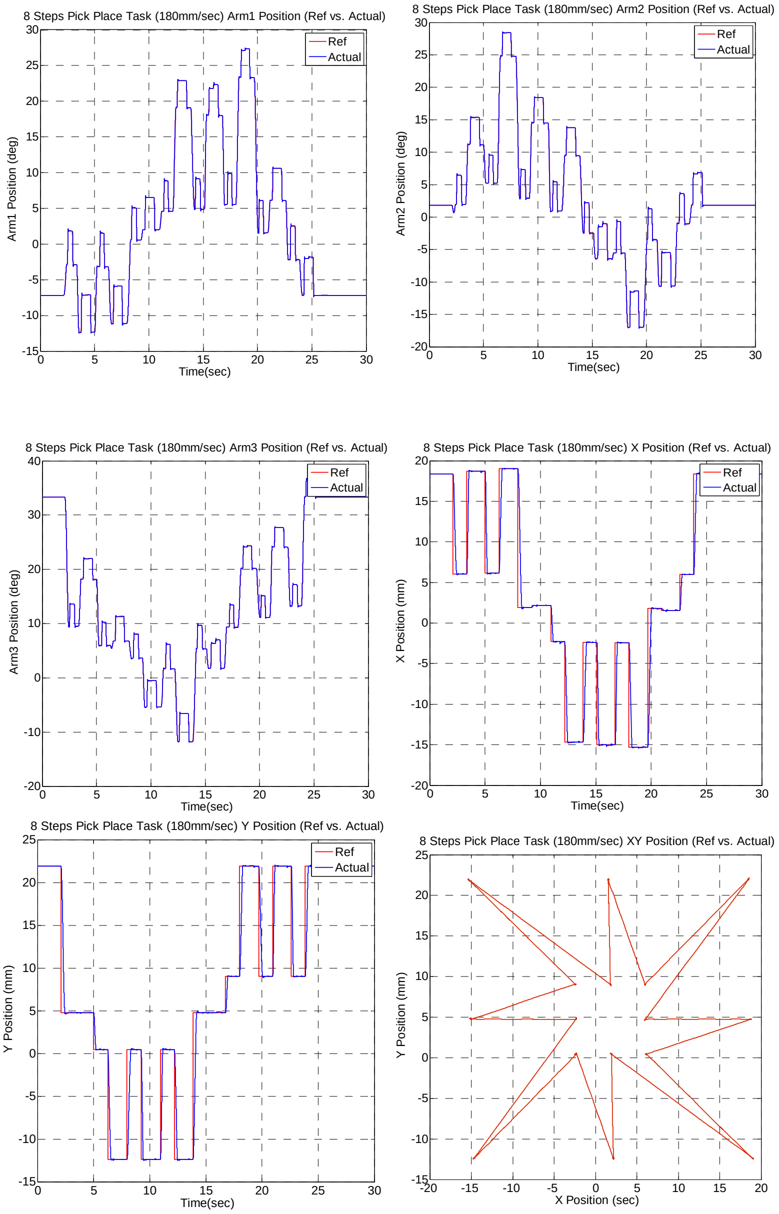


Figure 6-98 - 8 Steps Pick Place Experiment (180mm/sec Speed with ds1103 Controller Board)

Again the bottleneck for the cycle time of the task depends on the idle times for the vacuum operation. The idle times at pick and place positions should be optimized and determined according to the speed of the Delta robot and given as a predefined process parameter to the system.

The second level of modularity of the system is tested by adding a second Delta robot in line with the conveyor. The modular structure of the system and the task units allows easy integration of a second Delta robot. The robot can be integrated to the system within a couple of hours. The ease of integration of an additional task unit to the module is achieved by the modular structure of the module concept with all its features; software, electronics and hardware.

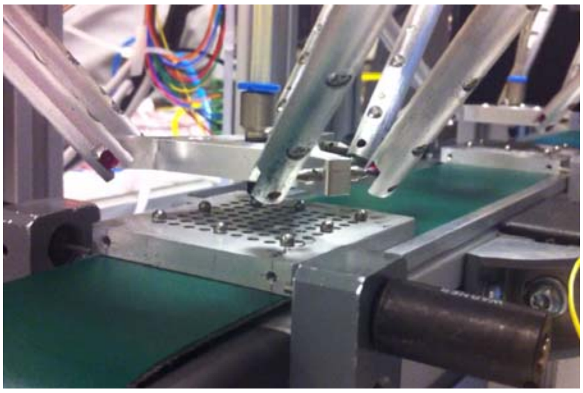


Figure 6-99 – Two Delta Robots Realizing the Assembly Operation Simultaneously

Addition of an extra manipulator makes the assembly system more versatile since it can enable the implementation of different assembly scenarios. With the integration of a different end effector tool to the robot makes it possible to manipulate different kind of parts. With this feature more complex assembly operations can be realized within the assembly module. On the other hand, the speed of the assembly process can be increased according to the implemented assembly scenario. Delta robots can work simultaneously and both of them realizing the same assembly process doubles the assembly speed.

The experiments realized with the addition of an extra task unit to the module are oriented towards testing the difference of the assembly speed of the system with the additional Delta robot. In order to make a reliable evaluation of the increase in the performance, the operator intervention should be minimized or even removed since time is the measure for the performance test and the duration of the operator intervention process (task generation) may change at each cycle. For that reason, an assembly procedure is defined as follows;

The parts are placed on the tray with the same position configuration during the whole assembly process. The assembly recipe (pick place tasks) is defined only once at the first cycle by the operator using the GUI after the inspection of the tray with the vision system and position determination of the parts. The operator realizes the assembly recipe as it was depicted in the explanation of the previous experiments. After the generation of the task, the system goes into an automatic cycle in which the operator is only responsible for filling the trays in the same configuration and feeding the trays on the conveyor continuously. Both Delta robots realize the same pick and place tasks simultaneously on the trays which are positioned at the workspace of the robots.

First of all, the parameters (idle times) that are used for the first experiments realized by using one Delta robot system are kept the same, results of which are given in the previous experiments realized with only one manipulator. The performance of the Delta robots is also aimed to be increased with a new S type trajectory generation algorithm. The idle times for pick place operations are kept the same and the maximum values for velocity and acceleration are defined to be 20mm/sec and 10mm/sec^2 respectively for the trajectory generation.

Figure 6-100 shows the actuated angle positions of the two Delta robots working simultaneously realizing the same assembly procedure defined by the operator.

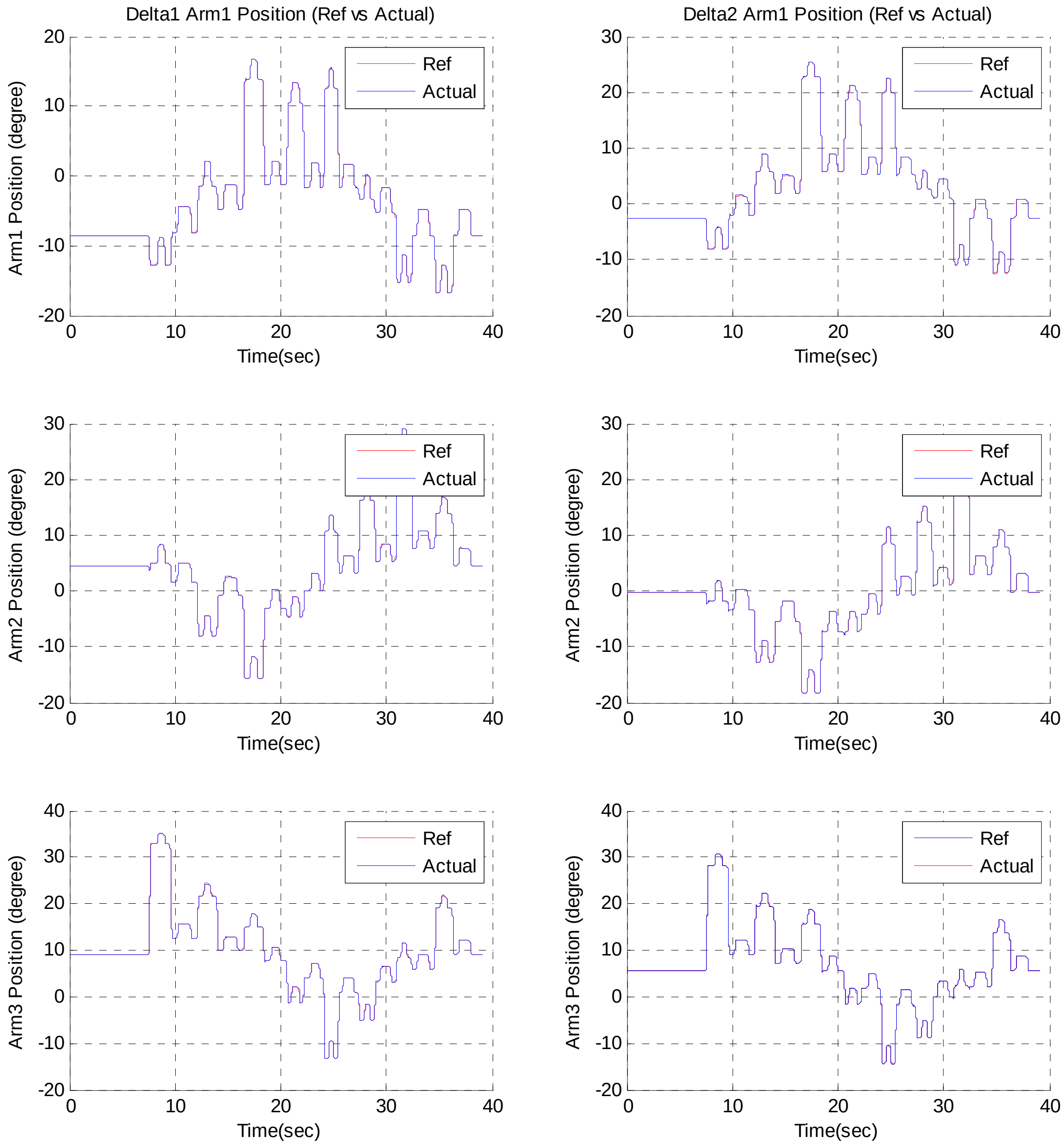


Figure 6-100 – Active Angle Positions for both Delta Robots for 20mm/sec and 10mm/sec^2 velocity and acceleration maximum values

The experiment realized is again 8 step pick place operation, this time implemented with two Delta robots. The end effector Cartesian coordinate positions of the Delta robots are given in Figure 6-101.

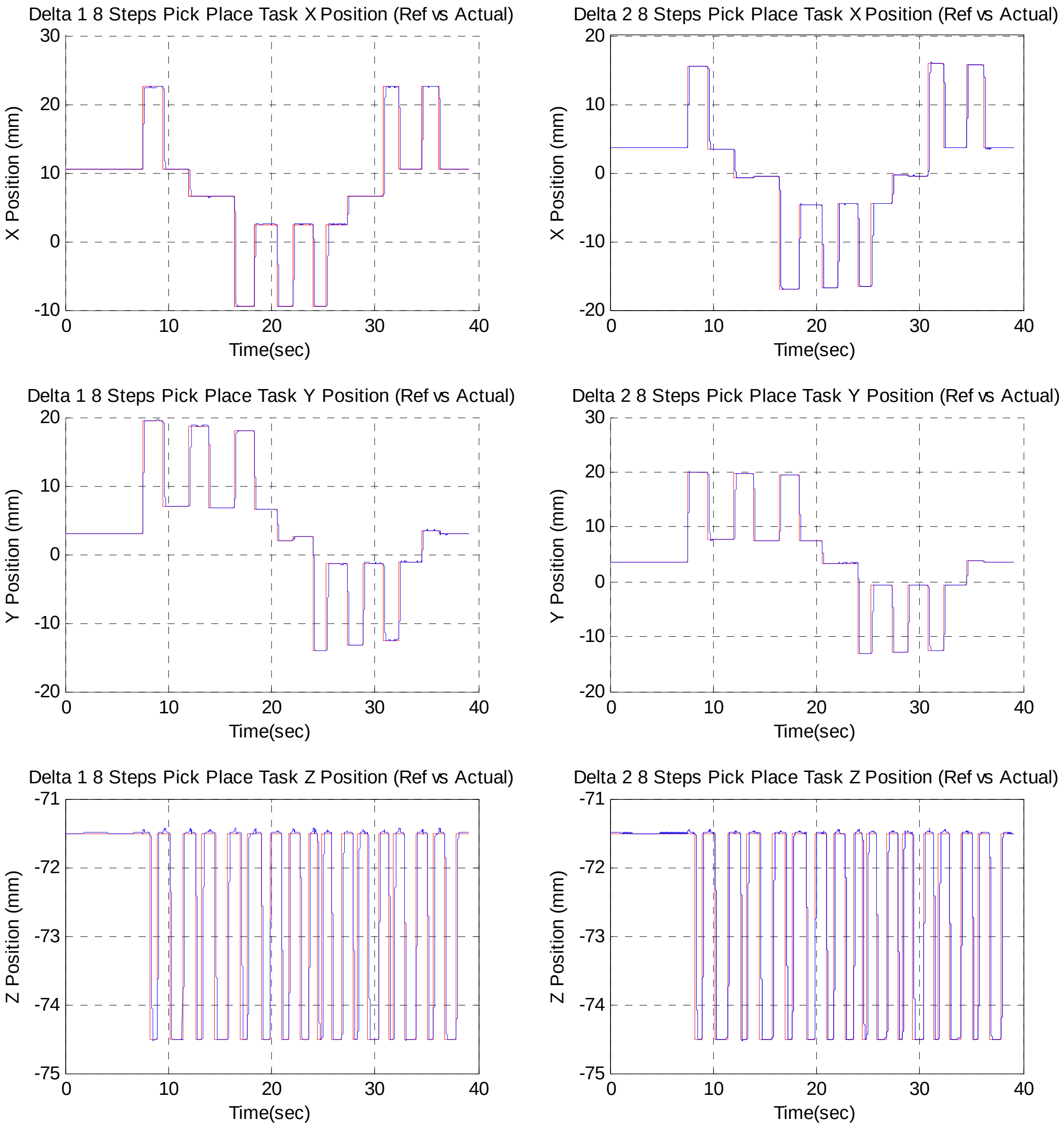


Figure 6-101 – X, Y and Z positions of the Delta robots for the 8 Step Pick Place Experiment

The assembly operation starts approximately at 7.52 seconds and finishes at 38.09 seconds. The whole assembly is finalized in 30.57 seconds. The process is realized at low speeds in order to check the simultaneous operation of the Delta robots. The parameters are gradually changed for the speed optimization process. The following figure shows the X and Y coordinates of the robots during the pick place operation.

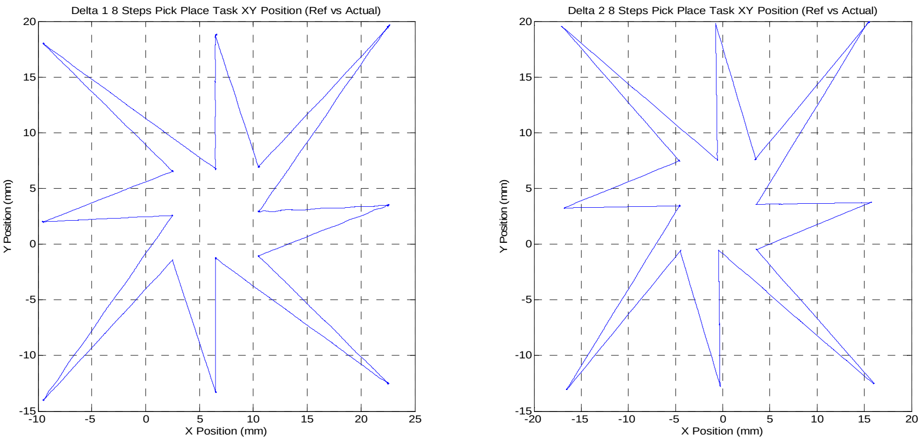
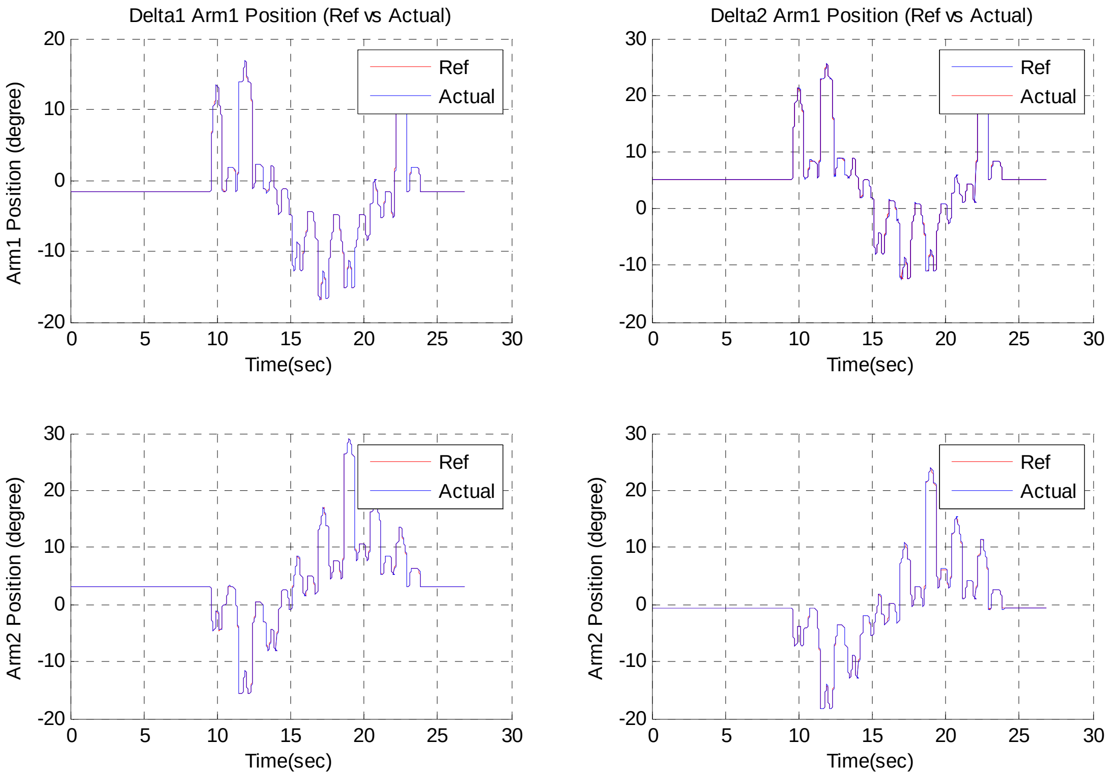


Figure 6-102 – X and Y Positions of the robots during the 8 Step Pick Place Operation

In order to increase the speed of the process, idle times for the vacuum on/off are reduced and the maximum velocity and acceleration values are set to 60 mm/sec and 20 mm/sec^2. The actuated angle positions and the X, Y and Z positions of the two Delta robots are given in Figure 6-103 and Figure 6-104 respectively. The experiment is realized again as 8 Step pick place operation, both Delta robots working simultaneously realizing the same assembly procedure. The motion starts at time 9.75 seconds and finalizes at time 23.87 seconds. The whole assembly operation lasts 14.12 seconds which approximately corresponds to 54 % of decrease in the duration of the assembly.



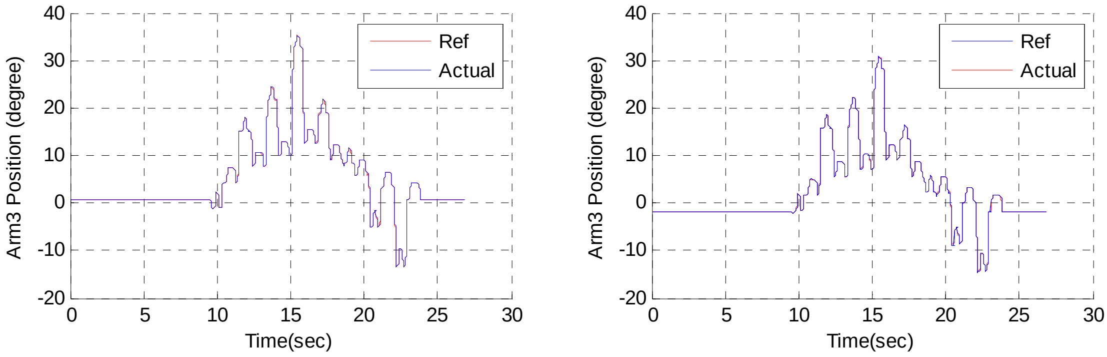


Figure 6-103 - Active Angle Positions for both Delta Robots for 20mm/sec and 10mm/sec^2 velocity and acceleration maximum values (Vel Max = 60mm/sec and Acc Max = 20 mm/sec^2)

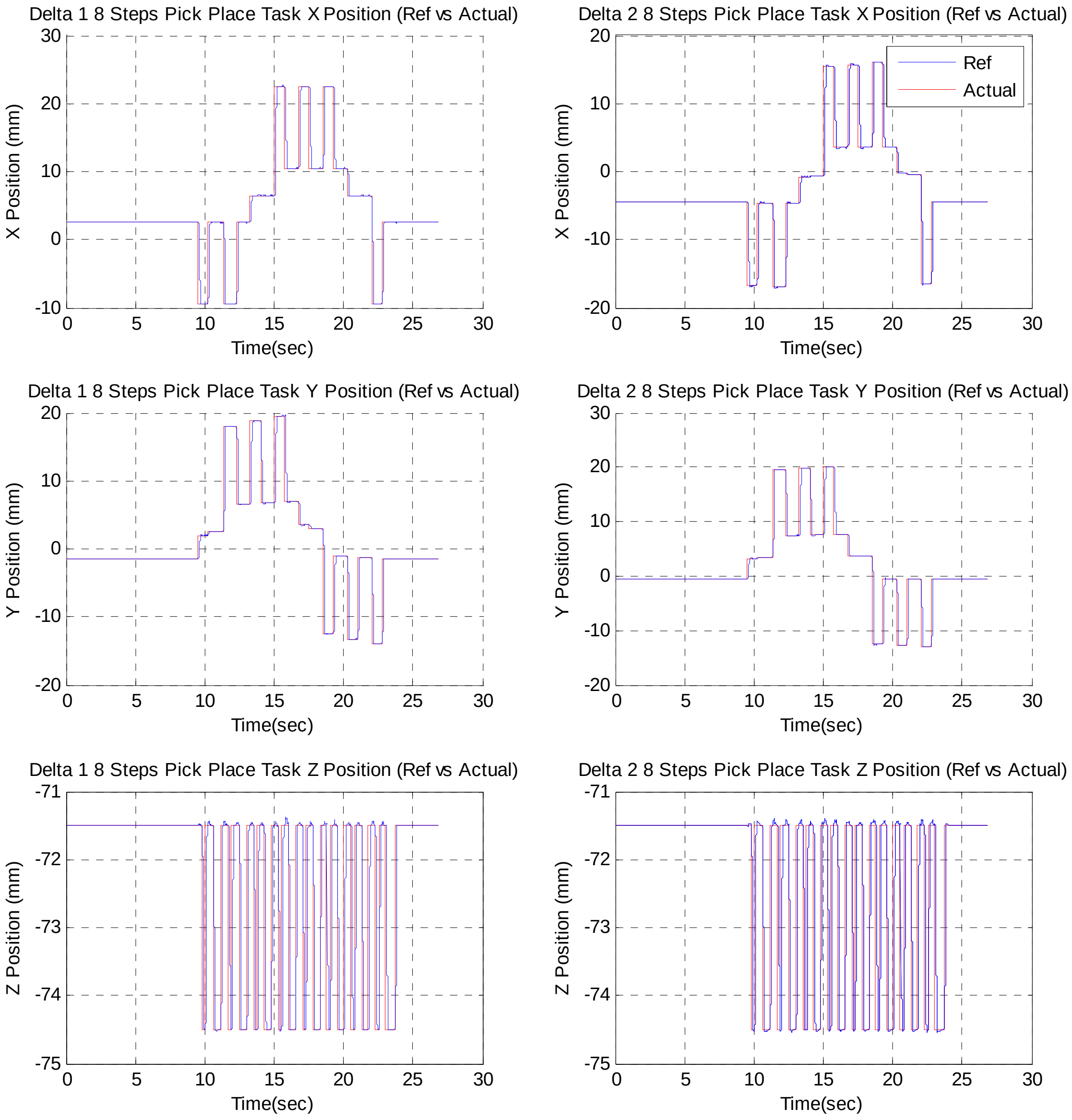


Figure 6-104 - X, Y and Z positions of the Delta robots for the 8 Step Pick Place Experiment (Vel Max = 60mm/sec and Acc Max = 20 mm/sec^2)

The following figure shows the X and Y coordinates of the robots during the pick place operation.

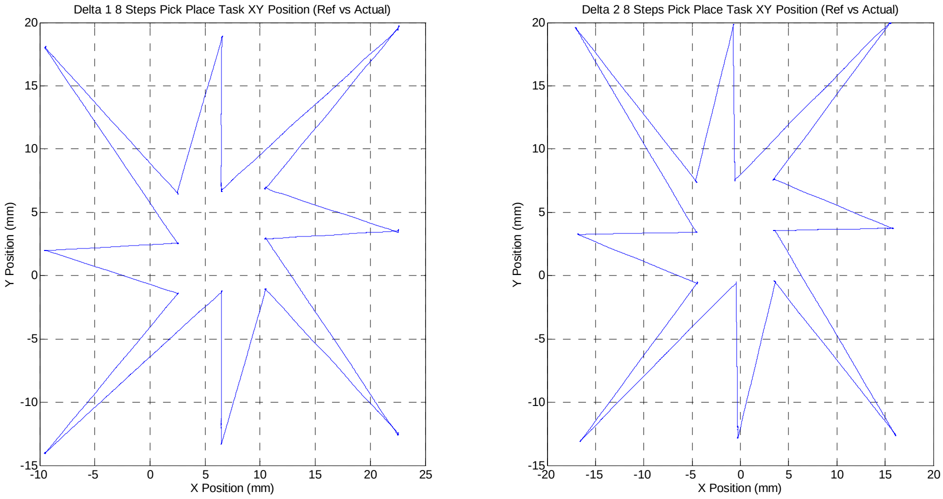
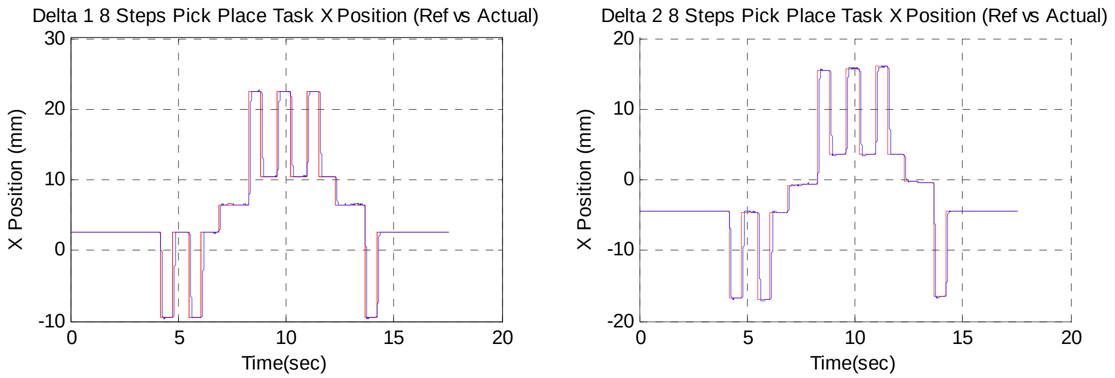


Figure 6-105 – X and Y Positions of the robots during the 8 Step Pick Place Operation (Vel Max = 60mm/sec and Acc Max = 20 mm/sec^2)

The idle times for the pick place operation are reduced and the maximum velocity and acceleration values for the reference generation are set to 100 mm/sec and 30mm /sec^2 in order to check whether it is possible to increase the speed performance of the assembly process or not. The motion approximately starts at 4.35 seconds and finishes 14.89 seconds, completing the whole assembly process in 10.54 seconds. When compared with the first and the second experiments there are 65.52% and 25.35% decrease in the duration of the assembly process.

The X,Y and Z positions of the two Delta robots are given with respect to time for the 8 steps pick place operation in Figure 6-106. In order to show the motion of the Delta robots on the XY plane to realize the process is given in Figure 6-107.



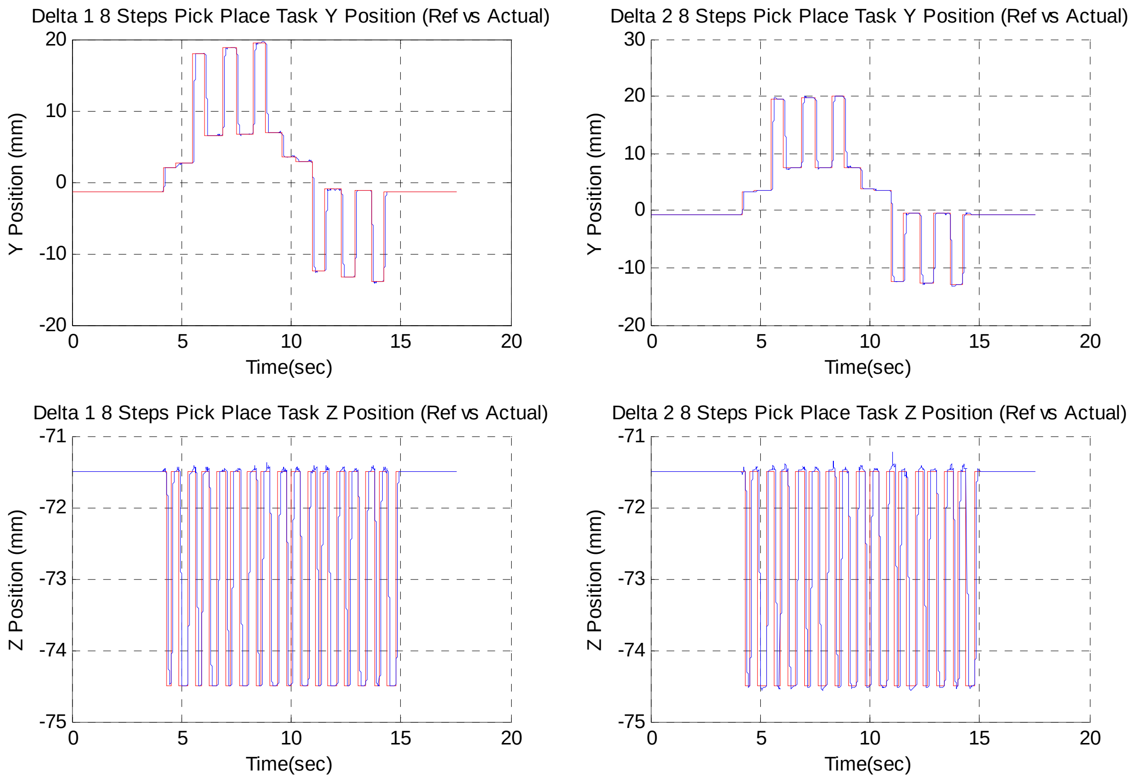


Figure 6-106 - X, Y and Z positions of the Delta robots for the 8 Step Pick Place Experiment (Vel Max = 100mm/sec and Acc Max = 30 mm/sec^2)

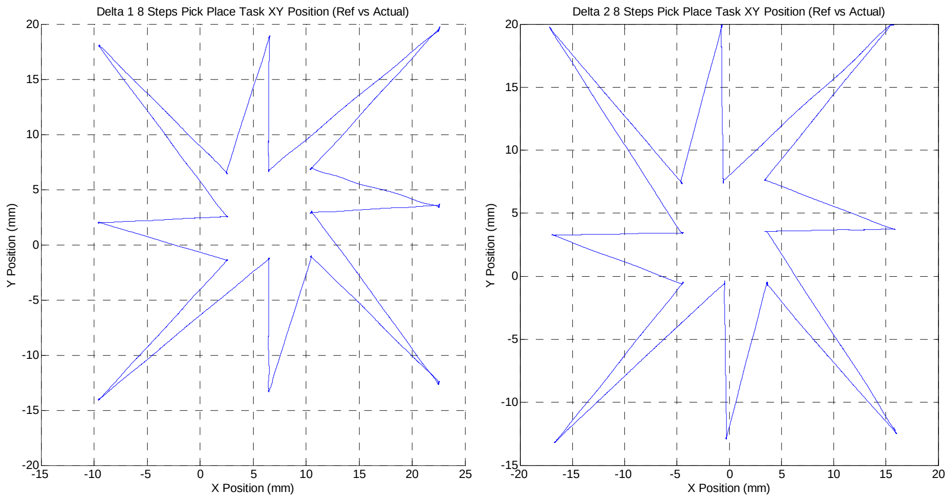


Figure 6-107 - X and Y Positions of the robots during the 8 Step Pick Place Operation (Vel Max = 100mm/sec and Acc Max = 30 mm/sec^2)

The system is tested at those speeds to realize the pick place operation. At each experiment the robots realize the pick and place tasks successfully.

6.4 Results and Discussion

The bilevel modularity concept is tested through the experiments realized with the different configuration of the system. The modularity within the module is proven by adding an additional robot in order to realize different tasks of assembly operations. By adding the second robot the system becomes more versatile for the assembly processes. That versatility can be described as the increase in the speed of operation as shown with experiments in the previous section. Other scenarios also become possible like assembly of different parts with the addition of different manipulation tools to the end effector of the robots. The experimental results show that the assembly operations are realized successfully for the targeted scenario of two manipulator system.

The ease of integration of an additional task unit to the process module is the most important part for the verification of the second level of modularity which is the modularity within the module. The Delta robot as the second manipulator is easily integrated to the system as a result of the modularity of the whole structure in both software and hardware.

With the optimization of the idle times and the trajectory generation with optimized maximum velocity and acceleration values, the speed of the process is also increased. The addition of an extra manipulator doubles the speed of the process with the defined scenario. In the final end, in 10.54 seconds 16 pick place operations can be realized within the assembly module not considering the only initial task generation part with the inspection system since it is realized only once at the beginning of a new assembly procedure. The results achieved are satisfactory in the sense of precision, repeatability and speed.

7 CONCLUSION

A microfactory concept with bilevel modularity is presented in this thesis work. The modularity concept which is inevitable in the concept of microfactories is enhanced by introducing the second level of modularity which is defined within the module. The first level of modularity is achieved with the design of the process oriented modules as a box type structure so that different modules can be cascaded easily to each other in order to form a more complex layout. Second level of modularity enables the easy configuration of the module within itself that brings out the customization of the modules according to the process necessities. In that sense, the modules with two layers of modularity which are defined to be POMs can have different task units within itself to realize a whole process.

The concept of bilevel modular microfactory concept is tested with the developed robotic assembly module and the experiments realized using the module demonstrated the advantages of the microfactory concept. Firstly, the assembly system is configured using an inspection system to determine the positions of the parts moving in between the stations by a conveyor unit and the assembly process defined as a pick place operation is realized by a single Delta robot as the manipulation unit. An additional Delta robot is integrated to the system without a major revision in the system structure and the same pick place experiments are realized by two Delta robots working simultaneously in the module. This addition of an extra robot doubles the speed of the system with the assembly scenario defined for the operation. It also brings versatility to the system since several assembly scenarios can be implemented in the module.

The task units used in the module are developed within the context of this thesis. Each manipulator is designed as a miniaturized system which is scaled in terms of dimensions considering the precision and accuracy requirements in order to realize high precision assembly tasks. Several experiments are realized using the manipulators before integrating into the system to test the performance parameters. The design of each system is evolved with the development of several prototypes till the satisfactory requirements are achieved.

The module concept realized within the context of this thesis forms a base platform for future integration of manufacturing modules such as laser micromachining unit and microassembly modules as a separate module integrated in the first level of modularity. On the other hand, the modularity structure gives possibility to the integration of the manufacturing task units into the module in the second level of modularity.

8 REFERENCES

[1] K. Furuta, “The Experimental Microfactory System in Japanese national R&D Project,” Singapore-Japan Forum on MEMS, 23 November 2000, Singapore.

[2] Fatikow, S.; Seyfried, J.; Fahlbusch, S.; Buerkle, A.; Schmoeckel, F., "A flexible microrobot-based microassembly station," *Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on* , vol.1, no., pp.397-406 vol.1, 1999

[3] Woern, H.; Seyfried, J.; St. Fahlbusch; Buerkle, A.; Schmoeckel, F., "Flexible microrobots for micro assembly tasks," *Micromechatronics and Human Science, 2000. MHS 2000. Proceedings of 2000 International Symposium on* , vol., no., pp.135-143, 2000

[4] Fatikow, S.; Rembold, U., "An automated microrobot-based desktop station for micro assembly and handling of micro-objects," *Emerging Technologies and Factory Automation, 1996. EFTA '96. Proceedings., 1996 IEEE Conference on*, vol.2, no., pp.586-592 vol.2, 18-21 Nov 1996.

[5] Dechev, N.; Lu Ren; Liu, W.; Cleghorn, L.; Mills, J.K.; , "Development of a 6 degree of freedom robotic micromanipulator for use in 3D MEMS microassembly," *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* , vol., no., pp.281-288, 15-19 May 2006

[6] Dechev, N.; Cleghorn, W.L.; Mills, J.K.; , "Microassembly of 3D MEMS structures utilizing a MEMS microgripper with a robotic manipulator," *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on* , vol.3, no., pp. 3193-3199 vol.3, 14-19 Sept. 2003

[7] Dechev, N.; Cleghorn, W.L.; Mills, J.K.; , "Microassembly of 3-D microstructures using a compliant, passive microgripper," *Microelectromechanical Systems, Journal of* , vol.13, no.2, pp. 176- 189, April 2004

[8] Anis, Y.H.; Mills, J.K.; Cleghorn, W.L.; , "Automated Microassembly Task Execution Using Vision-Based Feedback Control," *Information Acquisition, 2007. ICIA '07. International Conference on* , vol., no., pp.476-481, 8-11 July 2007

[9] Probst, M.; Vollmers, K.; Kratochvil, B. E.; Nelson, B. J.; , "Design of an Advanced Microassembly System for the Automated Assembly of Bio-Microrobots," in *Proceedings of 5th International Workshop on Microfactories*, 2004.

[10] Xinhan Huang; Xiadong Lv; Min Wang; , "Development of A Robotic Microassembly System with Multi-Manipulator Cooperation," *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on* , vol., no., pp.1197-1201, 25-28 June 2006

[11] Kunt, E.D.; Cakir, K.; Sabanovic, A.;, "A workstation for microassembly," *Control & Automation, 2007. MED '07. Mediterranean Conference on* , vol., no., pp.1-6, 27-29 June 2007

[12] Kunt, E.D.; Naskali, A.T.; Cakir, K.; Sabanovic, A.; Yuksel, E.; , "A Versatile and Reconfigurable Microassembly Workstation," *Proceedings of the International Workshop on Microfactories, 2008. IWMF'08,* pp. 37-41, October 2008.

[13] Kunt, E.D., "Design and Realization of a Microassembly Workstation", *MSc. Thesis*, 2006, Sabanci University, Istanbul, Turkey.

[14] Okazaki, Y.; Mishima, N.; Ashida, K.; , "Microfactory and Micro Machine Tools," *The 1st Korea-Japan Conference on Positioning Technology*, Daejeon, Korea, 2002

[15] Microfactory Prototype, October 2011. [online] http://www.olympus.co.jp/en/news/1999b/nr991201mifae.cfm

[16] Suda, M.; Furata, K.; Sakuhara, T.; Akata, T.; "The microfactory system using electrochemical machining", *Galvanotechnik Journal*, Japan, 2000, vol. 90, No 9, p. 2607-2609.

[17] Dobler, H.; Kuhn, C.; Klumpp, B.; , "Innovative Biochip Production Systems," *LabAutomation 2001*, January 27-31

[18] Dobler, H.; Kuhn, C.; Klumpp, B.; , “Innovative Solutions for biochip production systems”, *2nd International Workshop On Microfactories*, p. 115-118, October 2000.

[19] Gaugel, T.; Dobler, H., Nelson, B.J.; , “Advanced Modular Micro-Production System (AMMS),” *Proceedings of SPIE – The International Society for Optical Engineering 4568,* 2001, pp. 278-285.

[20] Gaugel, T.; Dobler, H.; Rohrmoser, B.; Klenk, J.; Neugebauer, J.; Schäfer, W.;, “Advanced modular production concept for miniaturized production”*, 2nd International Workshop On Microfactories*, *IWMF 2000,* pp. 35-38, October 2000.

[21] Hollis, R.L.; Quaid, A.; “An Architecture for Agile Assembly,” *Proc. Am. Soc. Of Precision Engineering,* October 1995.

[22] Lauwers, T.B.; Edmondson, Z.K.; Hollis, R.L.;, ”Progress in agile assembly: minifactory couriers based on free-roaming planar motors”, *4th international Workshop on Microfactories*, *IWMF 2004*, Vol. 1, p. 7-10, October 2004

[23] Heikkila, R.H.; Karjalainen, I.T.; Uusitalo, J.J.; Vuola, A.S.; Tuokko, R.O.; , "Possibilities of a Microfactory in the Assembly of Small Parts and Products - First Results of the M4-project," *Assembly and Manufacturing, 2007. ISAM '07. IEEE International Symposium on*, vol., no., pp.166-171, 22-25 July 2007

[24] Heikkilä, R; Huttunen, A; Vuola, A; Tuokko, R.;, “Microfactory concept for manufacturing of personalized medical implant”. *Proceedings of the 6th International Workshop on Microfactories IWMF 2008*, October 2008.

[25] Prusi, T.; Uusitalo, J. ; Heikkilä, R.; Tuokko, R.; , “Visual On-line Measurement in a Laser Micro Lathe,” *Proceedings of the 6th International Workshop on Microfactories IWMF 2008*, October 2008.

[26] Verettas, I.; Clavel, R.; Codourey, A.;, “PocketFactory : a modular and miniature assembly chain including a clean environment,” *Proceedings of the 6th International Workshop on MicroFactories IWMF 2006*, October 2006.

[27] Presentation on Microfactory, June 2005. [online] http://www.leti.cea.fr/commun/AR-2002/00-csem/02-claessen.pdf

[28] Vogler, M.P.; Liu, X.; Kapoor, S.G.; Devor, R.E.;, "Development of Mesoscale Machine Tool (mMt) Systems", *Transactions of the North American Manufacturing Research Institution of SME,* pp. 653-652, 2002.

[29] Mini Production System, December 2011. [online] http://www.sii.co.jp/info/eg/mini-fab1.html

[30] Ito, S.; Iijima, D.; Hayashi, A.; Aoyama, H.; Yamanaka, M.; "Micro turning system: A Super Small CNC Precision Lathe for Microfactories," *Proceedings of the 3rd International Workshop on Microfactories*, pp. 37–40, 2002.

[31] Stewart, D.; "A Platform with Six Degrees of Freedom: A new form of mechanical linkage which enables a platform to move simultaneously in all six degrees of freedom developed by Elliott-Automation," *Aircraft Engineering and Aerospace Technology*, Vol. 38 Iss: 4, pp.30 - 35

[32] Clavel, R.;, "Une nouvelle structure de manipulateur parallèle pour la robotique légère," *APII 23*, pp. 501-519, 1989.

[33] Pierrot, F.; Dauchez, P.; Fournier, A.; , "HEXA: a fast six-DOF fully-parallel robot," *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on* , vol., no., pp.1158-1163 vol.2, 19-22 Jun 1991

[34] Heikkila, R.H.; Karjalainen, I.T.; Uusitalo, J.J.; Vuola, A.S.; Tuokko, R.O.; , "Possibilities of a Microfactory in the Assembly of Small Parts and Products - First Results of the M4-project," *Assembly and Manufacturing, 2007. ISAM '07. IEEE International Symposium on* , vol., no., pp.166-171, 22-25 July 2007

[35] M-810 Miniature Hexapod, December 2011. [online] http://www.physikinstrumente.com/en/products/prdetail.php?sortnr=700880

[36] Hexapod HP-140, December 2011, [online] http://www.pimicos.com/web2/en/1,3,060,hp140.html

[37] Verettas, I.; Codourey, A., Clavel, R.: “Pocket Factory”: Concept of Miniaturized Modular Cleanrooms. *1st Topical Meeting on Microfactories “Desktop MEMS and Nanofactories” TMMF2005,* October 2005.

[38] Naskali, A.T.; “Software Framework for High Precision Motion Control Applications”, PhD Thesis, Ongoing, Sabanci University, Istanbul, Turkey

[39] Fitzgibbon, A.W.; , "Simultaneous linear estimation of multiple view geometry and lens distortion," *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* , vol.1, no., pp. I-125- I-132 vol.1, 2001

[40] Delta Robot, January 2010. [online] http://en.wikipedia.org/wiki/Delta_robot

[41] Laribi, A.B.; Romdhane, L.; Zeghloul, S.;, “Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace,” *Mechanism and Machine Theory*, vol. 42, pp. 859-870, July 2007.

[42] Miller, K., and Clavel, R., "The Lagrange-Based Model of Delta-4 Robot Dynamics," Robotersysteme, Springer-Verlag, Vol. 8, No. 4, pp. 49-54, 1992.

[43] Miller, K., "The Proposal of a New Model of Direct Drive Robot DELTA-4 Dynamics," *Int. Conf. on Advanced Robotics (ICAR'92),* Tokyo, Japan, pp. 411-416, November 1-2, 1992..

[44] Clavel, R. “Delta, A fast robot with parallel geometry,” *In Proceedings of the 18th International Symposium on Industrial Robots*, pp. 91–100, 1988,.

[45] Clavel, R., "Conception d'un robot parallèle rapide *à* 4 degrés *de* liberté," Ph.D. Thesis, EPFL, Lausanne, Switzerland, 1991

[46] Takase, K., “Task Oriented Variable Control of manipulator and Its Software Servoing System,” *Proc. IFAC Int. Symp.*, p.139, 1977.

[47] Khatib, O.; Le Maitre, J.F.; , “Dynamic Control of Manipulators Operating in a Complex Environment,” *Proc. 3rd CISM-IFToMM Symp. On Theory and Practice of Robots and Manipulators*, pp.267-282, 1978.

[48] Luh, J.; Walker, M.; Paul, R.; , "Resolved-acceleration control of mechanical manipulators," *Automatic Control, IEEE Transactions on* , vol.25, no.3, pp. 468- 474, Jun 1980

[49] Khatib, O.; , "A unified approach for motion and force control of robot manipulators: The operational space formulation," *Robotics and Automation, IEEE Journal of* , vol.3, no.1, pp.43-53, February 1987

[50] Khatib, 0., "Inertial Properties in Robotic Manipulation: An Object-Level Framework"*, International Journal of Robotics Research*, Vol 14, No. 1, February 1995, pp 19-36..

[51] Sapio, V.; Khatib, O.; Delp, S.; , “Task Level Approaches for the Control of Constrained Multibody Systems,” *Multibody System Dynamics,* vol. 16, no:1, pp.73-102, 2006

[52] Codourey, A.; Clavel, R.; Burckhardt, C.W.; "Control algorithm and controller for the Direct Drive Delta Robot", *in Proc. IFAC Symposium on robot Control,* pp. 169-177, 1991.

[53] Miller, K., "Modeling of Dynamics and Model-Based Control of DELTA Direct-Drive Parallel Robot," Journal of Robotics and Mechatronics, Vol. 17, No. 4, pp. 344-352, 1995.

[54] Codourey, A., "Dynamic modelling and mass matrix evaluation of the DELTA parallel robot for axes decoupling control," *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on* , vol.3, no., pp.1211-1218 vol.3, 4-8 Nov 1996

[55] Pierrot, F.; Fournier, A.; Dauchex, P., "Towards a fully-parallel 6 DOF robot for high-speed applications ," *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on* , vol., no., pp.1288-1293 vol.2, 9-11 Apr 1991.

[56] Pierror F.; Dauchez P.; Fournier A.;, "Fast Parallel Robots," *Journal of Robotic Systems*, vol.8, pp. 829-840, 1991.

[57] Codourey, A.; Burdet, E., "A body-oriented method for finding a linear form of the dynamic equation of fully parallel robots," *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on* , vol.2, no., pp.1612-1618 vol.2, 20-25 Apr 1997.

[58] Kane T. R. and Levinson D.A., Dynamics: Theory and Applications, McGraw-Hill, New York, 1985

[59] Spong, M.W.; Vidyasagar, M; , "Robot Dynamics and Control," John Wiley and Sons, New York, 1989.

[60] Ramstein, C.; Hayward, V.; "The pantograph: a large workspace haptic device for multimodal human computer interaction", *in Proc. CHI Conference Companion*, pp.57-58, 1994.

[61] Campion, G.; Qi Wang; Hayward, V.; , "The Pantograph Mk-II: a haptic instrument," *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on* , vol., no., pp. 193- 198, 2-6 Aug. 2005

[62] Ohnishi, K.; Shibata, M.; Murakami, T.; , "Motion Control for Advanced Mechatronics", *ASME Transaction On Mechatronics*, vol. 1, no. 1, pp. 56-67, March. 1996.

[63] Khalil, I. S. M.; Golubovic, E.; Sabanovic, A.; , "High precision motion control of parallel robots with imperfections and manufacturing tolerances," *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on* , vol., no., pp.1-7, 13-16 Sept. 2010

[64] Murakami, T.; Ohnishi, K.; , "Observer-based motion control-application to robust control and parameter identification," *Motion Control Proceedings, 1993., Asia-Pacific Workshop on Advances in* , vol., no., pp.1-6, 15-16 Jul 1993

[65] Murakami, T.; Ohnishi, K.; , ”Robust And Adaptive Control Strategies,” *in Proc In. Conf. IFAC Motion Control for Intelligent Automation.*, pp. 367-372, October. 1992.

[66] Katsura, S.; Ohnishi, K.; , "Force Servoing by Flexible Manipulator Based on Resonance Ratio Control," *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on* , vol.4, no., pp. 1343- 1348, June 20-23, 2005.

[67] Cortesao, R.; Zarrad, W.; Poignet, P.; Company, O.; Dombre, E.; , ”Task-Space and Null-Space Control Design for Robotic-Assisted Minimally Invasive Surgery,” *in Proc Int. Conf. IROS06 International Conference Intelligent Robots and Systems.*, pp. 454-459, 2006.

[68] Khan, S.; Elitas, M.; Kunt, E.D.; Sabanovic, A.; , "Discrete Sliding Mode Control of Piezo Actuator in Nano-Scale Range," *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on* , vol., no., pp.1454-1459, 15-17 Dec. 2006.

APPENDIX

A.1) Inverse Kinematics of Delta Robot (C Code)

```
Float64 LA = 40;
Float64 LB = 68;
Float64 RA = 40.50;
Float64 RB = 30;
Float64 Theta1 = 0*pi/180;
Float64 Theta2 = 120*pi/180;
Float64 Theta3 = 240*pi/180;

void Delta_Inv(Float64* x,
                    Float64* y,
                    Float64* z,
                    Float64* alpha1,
                    Float64* alpha2,
                    Float64* alpha3)
{
Float64 R = RA-RB;
Float64 z_o = *z;
Float64 x_2 = pow(*x,2);
Float64 y_2 = pow(*y,2);
Float64 z_o_2 = pow(z_o,2);
Float64 LA_2 = pow(LA,2);
Float64 LB_2 = pow(LB,2);
Float64 R_2 = pow(R,2);

Float64 S = (1/LA)*(-x_2-y_2-z_o_2+LB_2-LA_2-R_2);
Float64 Q1 = 2*(*x)*cos(Theta1)+2*(*y)*sin(Theta1);
Float64 Q2 = 2*(*x)*cos(Theta2)+2*(*y)*sin(Theta2);
Float64 Q3 = 2*(*x)*cos(Theta3)+2*(*y)*sin(Theta3);

Float64 Q1_2 = pow(Q1,2);
Float64 Q2_2 = pow(Q2,2);
Float64 Q3_2 = pow(Q3,2);

Float64 S_2 = pow(S,2);

Float64  T1  =  (2*z_o+sqrt(4*z_o_2+4*R_2-S_2+Q1_2*(1-R_2/LA_2)+(Q1)*(-
2*R*(S)/LA-4*R)))/(-2*R-(S)-(Q1)*(R/LA-1));
Float64  T2  =  (2*z_o+sqrt(4*z_o_2+4*R_2-S_2+Q2_2*(1-R_2/LA_2)+(Q2)*(-
2*R*(S)/LA-4*R)))/(-2*R-(S)-(Q2)*(R/LA-1));
Float64  T3  =  (2*z_o+sqrt(4*z_o_2+4*R_2-S_2+Q3_2*(1-R_2/LA_2)+(Q3)*(-
2*R*(S)/LA-4*R)))/(-2*R-(S)-(Q3)*(R/LA-1));

*alpha1 = (180/pi)*(-2*atan(T1)) - 30.0; // the offset is the initial
condition of
*alpha2 = (180/pi)*(-2*atan(T2)) - 30.0; //39.935;  // the motors.
*alpha3 = (180/pi)*(-2*atan(T3)) - 30.0; //39.935;
}
```

A.2) Forward Kinematics of Delta Robot (C Code)

```
void Delta_Fwd( Float64 alpha1,
                        Float64 alpha2,
                        Float64 alpha3,
                        Float64* x_out,
                        Float64* y_out,
                        Float64* z_out)
{
Float64 alpha1rad = (alpha1 + 30.0)*(pi/180);
Float64 alpha2rad = (alpha2 + 30.0)*(pi/180);
Float64 alpha3rad = (alpha3 + 30.0)*(pi/180);
Float64 R = RA-RB;
Float64 LA_2 = pow(LA,2);
Float64 LB_2 = pow(LB,2);
Float64 R_2 = pow(R,2);

Float64 D1 = -LB_2 + LA_2 + R_2 + 2*R*LA*cos(alpha1rad);
Float64 E1 = 2*(R+LA*cos(alpha1rad))*cos(Theta1);
Float64 F1 = 2*(R+LA*cos(alpha1rad))*sin(Theta1);
Float64 G1 = -2*LA*sin(alpha1rad);

Float64 D2 = -LB_2 + LA_2 + R_2 + 2*R*LA*cos(alpha2rad);
Float64 E2 = 2*(R+LA*cos(alpha2rad))*cos(Theta2);
Float64 F2 = 2*(R+LA*cos(alpha2rad))*sin(Theta2);
Float64 G2 = -2*LA*sin(alpha2rad);

Float64 D3 = -LB_2 + LA_2 + R_2 + 2*R*LA*cos(alpha3rad);
Float64 E3 = 2*(R+LA*cos(alpha3rad))*cos(Theta3);
Float64 F3 = 2*(R+LA*cos(alpha3rad))*sin(Theta3);
Float64 G3 = -2*LA*sin(alpha3rad);

Float64 H1 = E1*G2 - E1*G3 - E2*G1 + E2*G3 + E3*G1 - E3*G2;
Float64 H2 = -E1*F2 + E1*F3 + E2*F1 - E2*F3 - E3*F1 + E3*F2;
Float64 H3 = -E1*D2 + E1*D3 + E2*D1 - E2*D3 - E3*D1 + E3*D2;
Float64 H4 = F1*D2 - F1*D3 - F2*D1 + F2*D3 + F3*D1 - F3*D2;
Float64 H5 = -F1*G2 + F1*G3 + F2*G1 - F2*G3 - F3*G1 + F3*G2;

Float64 H1_2 = pow(H1,2);
Float64 H2_2 = pow(H2,2);
Float64 H3_2 = pow(H3,2);
Float64 H4_2 = pow(H4,2);
Float64 H5_2 = pow(H5,2);

Float64 L = ((H5_2 + H1_2)/H2_2) + 1;
Float64 M = 2*((H5*H4 + H1*H3)/H2_2) - ((H5*E1 + H1*F1)/H2) - G1;
Float64 Q = ((H4_2 + H3_2)/H2_2) - ((H4*E1 + H3*F1)/H2) + D1;

Float64 M_2 = pow(M,2);

*z_out = (-M - sqrt(M_2 - 4*L*Q))/(2*L);
*x_out = (*z_out*H5)/H2 + H4/H2;
*y_out = (*z_out*H1)/H2 + H3/H2;
}
```

A.3) AUTOLEV Code for Delta Robot Dynamics

```
%----------------------------------------------------------
%  Author:   Emrah Deniz Kunt
%  Problem:  Dynamics w/ Kane Method

%----------------------------------------------------------
% Default Settings
%----------------------------------------------------------
AutoEpsilon 1.0E-8 % Rounds off to the nearest integer
AutoZ       ON         % Turn ON for large problems
Digits       7         % Number of digits displayed for numbers

%----------------------------------------------------------
% Newtonian, bodies, frames, particles, points
%----------------------------------------------------------
Newtonian N
% A-D-G Upper arms C-F-I parallelograms
Bodies      A,C,D,F,G,I,K % K--> Nacelle
frames      B,E,H,P,R
Points      O,Z,Sh{3},El{3},Wr{3}

%----------------------------------------------------------
% Variables, constants, specified
%----------------------------------------------------------
Specified TOA, TOD, TOG
Specified   Fx,Fy,Fz
Variables alpha{3}',beta{3}',gamma{3}',x',y',z'
Constants La, Lb, Ra, Rb, Layoff, Laxoff, Zoff
Constants g = 9.81  % Local Gravitational Acceleration

%----------------------------------------------------------
% Motion Variables for static,dynamic analysis
%----------------------------------------------------------
Motionvariables'   u{12}'    % Motion variables; derivatives - here you
declare the whole set, including the Config ones

%----------------------------------------------------------
% Mass and Inertia
%----------------------------------------------------------
Mass     A=m1, C=m2, D=m1, F=m2, G=m1, I=m2, K=m7

Inertia  A,IA11, IA22, IA33, IA12, IA23, IA31
Inertia  C,IC11, IC22, IC33, IC12, IC23, IC31
Inertia  D,IA11, IA22, IA33, IA12, IA23, IA31
Inertia  F,IC11, IC22, IC33, IC12, IC23, IC31
Inertia  G,IA11, IA22, IA33, IA12, IA23, IA31
Inertia  I,IC11, IC22, IC33, IC12, IC23, IC31
Inertia  K,IK11, IK22, IK33, IK12, IK23, IK31

%----------------------------------------------------------
% Geometry relating unit vectors
%----------------------------------------------------------
Simprot(N,A,1,-alpha1)
Dircos(A,C,BODY132,-beta1,gamma1,0)

Simprot(N,R,3,2*pi/3)
Simprot(R,D,1,-alpha2)
Dircos(D,F,BODY132,-beta2,gamma2,0)

Simprot(N,P,3,4*pi/3)
Simprot(P,G,1,-alpha3)
Dircos(G,I,BODY132,-beta3,gamma3,0)

Dircos(N,K,BODY123,0,0,0)

%----------------------------------------------------------
% Angular Velocities
%----------------------------------------------------------
ANGVEL(N,R)
ANGVEL(N,P)

ANGVEL(N,A)
ANGVEL(N,D)
ANGVEL(N,G)

ANGVEL(N,C)
ANGVEL(N,F)
ANGVEL(N,I)

ANGVEL(N,K)

%----------------------------------------------------------
% Kinematical Differential Equations
%----------------------------------------------------------
alpha1' = u1
alpha2' = u2
alpha3' = u3
beta1'  = u4
beta2'  = u5
beta3'  = u6
gamma1' = u7
gamma2' = u8
gamma3' = u9
x' = u10
y' = u11
z' = u12

%----------------------------------------------------------
% Angular Accelerations
%----------------------------------------------------------
Alf_R_N> = dt(W_R_N>,N)
Alf_P_N> = dt(W_P_N>,N)

Alf_A_N> = dt(W_A_N>,N)
Alf_D_N> = dt(W_D_N>,N)
Alf_G_N> = dt(W_G_N>,N)

Alf_C_N> = dt(W_C_N>,N)
Alf_F_N> = dt(W_F_N>,N)
Alf_I_N> = dt(W_I_N>,N)

Alf_K_N> = dt(W_K_N>,N)

%----------------------------------------------------------
% Position Vectors (defined to the center of mass)
%----------------------------------------------------------
P_O_O>  = 0>

% Position vectors to the joints

P_O_Sh1>   = Ra*n2>
```

```
P_O_Sh2>   = Ra*r2>
P_O_Sh3>   = Ra*p2>

P_Sh1_El1> = La*a2>
P_Sh2_El2> = La*d2>
P_Sh3_El3> = La*g2>

P_El1_Wr1> = Lb*c2>
P_El2_Wr2> = Lb*f2>
P_El3_Wr3> = Lb*i2>

P_Wr1_Z> = -Rb*n2>
P_Wr2_Z> = -Rb*r2>
P_Wr3_Z> = -Rb*p2>

P_O_Z>  = -x*n1>-y*n2>-z*n3>

% Position vectors to the center of mass of bodies

P_O_Ao> = P_O_Sh1> + Layoff*a2> + Laxoff*a1>
P_O_Co> = P_O_Sh1> + P_Sh1_El1> + (0.5*P_El1_Wr1>)

P_O_Do> = P_O_Sh2> + Layoff*d2> + Laxoff*d1>
P_O_Fo> = P_O_Sh2> + P_Sh2_El2> + (0.5*P_El2_Wr2>)

P_O_Go> = P_O_Sh3> + Layoff*g2> + Laxoff*g1>
P_O_Io> = P_O_Sh3> + P_Sh3_El3> + (0.5*P_El3_Wr3>)

P_O_Ko> = P_O_Sh1> + P_Sh1_El1> + P_El1_Wr1> + P_Wr1_Z> + Zoff*n3>
% Zoff = 0.47

%----------------------------------------------------------
% Velocities
%----------------------------------------------------------
V_O_N>  = dt(P_O_O>,N)

V_Sh1_N> = dt(P_O_Sh1>,N)
V_Sh2_N> = dt(P_O_Sh2>,N)
V_Sh3_N> = dt(P_O_Sh3>,N)

V_El1_N> = dt(P_O_El1>,N)
V_El2_N> = dt(P_O_El2>,N)
V_El3_N> = dt(P_O_El3>,N)

V_Wr1_N> = dt(P_O_Wr1>,N)
V_Wr2_N> = dt(P_O_Wr2>,N)
V_Wr3_N> = dt(P_O_Wr3>,N)

V_Z_N>  = dt(P_O_Z>,N)

V_Ao_N> = dt(P_O_Ao>,N)
V_Co_N> = dt(P_O_Co>,N)

V_Do_N> = dt(P_O_Do>,N)
V_Fo_N> = dt(P_O_Fo>,N)

V_Go_N> = dt(P_O_Go>,N)
V_Io_N> = dt(P_O_Io>,N)

V_Ko_N> = dt(P_O_Ko>,N)
%----------------------------------------------------------
```

```
% Accelerations
%----------------------------------------------------------
A_O_N>  = dt(V_O_N>,N)

A_Sh1_N> = dt(V_Sh1_N>,N)
A_Sh2_N> = dt(V_Sh2_N>,N)
A_Sh3_N> = dt(V_Sh3_N>,N)

A_El1_N> = dt(V_El1_N>,N)
A_El2_N> = dt(V_El2_N>,N)
A_El3_N> = dt(V_El3_N>,N)

A_Wr1_N> = dt(V_Wr1_N>,N)
A_Wr2_N> = dt(V_Wr2_N>,N)
A_Wr3_N> = dt(V_Wr3_N>,N)

A_Z_N>  = dt(V_Z_N>,N)

A_Ao_N> = dt(V_Ao_N>,N)
A_Co_N> = dt(V_Co_N>,N)

A_Do_N> = dt(V_Do_N>,N)
A_Fo_N> = dt(V_Fo_N>,N)

A_Go_N> = dt(V_Go_N>,N)
A_Io_N> = dt(V_Io_N>,N)

A_Ko_N> = dt(V_Ko_N>,N)

%----------------------------------------------------------
% Configuration Constraints
%----------------------------------------------------------
Loop1>  = P_O_Sh1>  + P_Sh1_El1> + P_El1_Wr1> + P_Wr1_Z> - P_O_Z>
Loop2>  = P_O_Sh2>  + P_Sh2_El2> + P_El2_Wr2> + P_Wr2_Z> - P_O_Z>
Loop3>  = P_O_Sh3>  + P_Sh3_El3> + P_El3_Wr3> + P_Wr3_Z> - P_O_Z>

Config[1]=dot(Loop1>,N1>)
Config[2]=dot(Loop2>,N1>)
Config[3]=dot(Loop3>,N1>)
Config[4]=dot(Loop1>,N2>)
Config[5]=dot(Loop2>,N2>)
Config[6]=dot(Loop3>,N2>)
Config[7]=dot(Loop1>,N3>)
Config[8]=dot(Loop2>,N3>)
Config[9]=dot(Loop3>,N3>)

%----------------------------------------------------------
% Motion Constraints
%----------------------------------------------------------
dLoop1> = dt(Loop1>,N)
dLoop2> = dt(Loop2>,N)
dLoop3> = dt(Loop3>,N)

Dependent[1]=dot(dLoop1>,N1>)
Dependent[2]=dot(dLoop2>,N1>)
Dependent[3]=dot(dLoop3>,N1>)
Dependent[4]=dot(dLoop1>,N2>)
Dependent[5]=dot(dLoop2>,N2>)
Dependent[6]=dot(dLoop3>,N2>)
Dependent[7]=dot(dLoop1>,N3>)
Dependent[8]=dot(dLoop2>,N3>)
```

```
Dependent[9]=dot(dLoop3>,N3>)

Constrain(Dependent[u4,u5,u6,u7,u8,u9,u10,u11,u12])

%---------------------------------------------------------
% Jacobian
%---------------------------------------------------------
J = [coef(u10, u1), coef(u10, u2), coef(u10, u3); &
     coef(u11, u1), coef(u11, u2), coef(u11, u3); &
     coef(u12, u1), coef(u12, u2), coef(u12, u3)]

%---------------------------------------------------------
% Forces
%---------------------------------------------------------
Gravity(-g*n3>)
Force_Z> = Fx*n1>+Fy*n2>+Fz*n3>

Torque_A> = -TOA*a1>
Torque_D> = -TOD*d1>
Torque_G> = -TOG*g1>

%---------------------------------------------------------
% Units System for CODE input/output conversions
%---------------------------------------------------------
UnitSystem g,millimeter,sec
%-----------------------------------------------------------

Zero = Fr() + FrStar()
Kane()

Solve(Zero, [u1', u2', u3'])
M= -[COEF(Zero,u1'), COEF(Zero,u2'),COEF(Zero,u3')]
CG = -(Zero + M*[u1';u2';u3'])

Mtr = transpose(M)
Minv = inv(M)
Minvtr = inv(Mtr)

Encode  M, Mtr, Minv, Minvtr, CG

code dynamics() Delta_kane.m
code dynamics() Delta_kane.c
```

A.4) Dynamics of Delta Robot (C Code)

The following function is given as an example for the implementation of the Delta robot functional blocks. It is the C code generated to use for the dynamics using Kane’s method. In order to ease the calculations Z functions are used for the calculations and that part is omitted from the given code in order to save space.

```
void Delta_kane    (const double* Fext,  const double* Tctrl,
                   const double* pos, const double* vel, double* acc)
{
```

```
double Z[870] = {0};
const double pi = 3.14159265358979323846264338327950;

// Configuration Variable List
 //-----------------------------------------------------------------
---------------------------------
double        alpha1 = pos[0], alpha2 = pos[1], alpha3 = pos[2];
// Independent coordinates
double    beta1 = pos[3], beta2 = pos[4], beta3 = pos[5];
double    gamma1 = pos[6], gamma2 = pos[7], gamma3 = pos[8];
double    x = pos[9], y = pos[10], z = pos[11];

double    u1 = vel[0], u2 = vel[1], u3 = vel[2];
double    u10, u11, u12, u4, u5, u6, u7, u8, u9, alpha1p, alpha2p,
          alpha3p, beta1p, beta2p, beta3p, gamma1p, gamma2p, gamma3p,
          u1p, u2p, u3p, xp, yp, zp;

double    La =40,  Lb = 68 , Ra=40, Rb=30; // mm
double    m1 = 8.94, m2 = 18.51, m7 = 29.40; //grams
double    g = 9810.0; //mm/s^2

double    IA11=2321.70, IC11=6925.51, IC12=0.0, IC22=4385.78,
          IC23=0.01, IC31=0.0, IC33=11158.06; //gr/mm^2

/* Dynamic Parameters */
double    Fx = Fext[0], Fy = Fext[1], Fz = Fext[2];      // Forces on
the endeffector
double    TOA = Tctrl[0], TOD = Tctrl[1], TOG = Tctrl[2];// Torques on
the joints

/* Evaluate constants */
alpha1p = u1;
alpha2p = u2;
alpha3p = u3;
…….
…….
…….
//Z Function Calculations
…….
……..
……..
acc[0] = u1p;
acc[1] = u2p;
acc[2] = u3p;
}
```

A.5) S-Function of Dynamics of Delta Robot

```
#define S_FUNCTION_NAME  sfun_Delta_kane
#define S_FUNCTION_LEVEL 2

#include "simstruc.h"
#include <math.h>


/*====================*
 * S-function methods *
 *====================*/
```

```
extern void  Delta_kane (const double* Fext,  const double* Tctrl,
                  const double* pos, const double* vel, double* acc);

/* Function: mdlInitializeSizes
===============================================
* The sizes information is used by Simulink to determine the S-
function block's characteristics (number of inputs, outputs, states,
etc.).*/

static void mdlInitializeSizes(SimStruct *S)
{
    /* See sfuntmpl_doc.c for more details on the macros below */
    ssSetNumSFcnParams(S, 0);  /* Number of expected parameters */
    if (ssGetNumSFcnParams(S) != ssGetSFcnParamsCount(S)) {
    /* Return if number of expected != number of actual parameters */
    return;
}

ssSetNumContStates(S, 0);
ssSetNumDiscStates(S, 0);

if (!ssSetNumInputPorts(S, 4)) return;

/*Configure the input port 1 -- fextF*/
ssSetInputPortDataType(S, 0, SS_DOUBLE);
ssSetInputPortWidth(S, 0, 3);
ssSetInputPortComplexSignal(S, 0, COMPLEX_NO);
ssSetInputPortRequiredContiguous(S, 0, true);
ssSetInputPortDirectFeedThrough(S, 0, 1);
ssSetInputPortOptimOpts(S, 0, SS_REUSABLE_AND_LOCAL);

/*Configure the input port 2 -- Tctrl*/
ssSetInputPortDataType(S, 1, SS_DOUBLE);
ssSetInputPortWidth(S, 1, 3);
ssSetInputPortComplexSignal(S, 1, COMPLEX_NO);
ssSetInputPortRequiredContiguous(S, 1, true);
ssSetInputPortDirectFeedThrough(S, 1, 1);
ssSetInputPortOptimOpts(S, 1, SS_REUSABLE_AND_LOCAL);

/*Configure the input port 3 -- pos*/
ssSetInputPortDataType(S, 2, SS_DOUBLE);
ssSetInputPortWidth(S, 2, 12);
ssSetInputPortComplexSignal(S, 2, COMPLEX_NO);
ssSetInputPortRequiredContiguous(S, 2, true);
ssSetInputPortDirectFeedThrough(S, 2, 1);
ssSetInputPortOptimOpts(S, 2, SS_REUSABLE_AND_LOCAL);

/*Configure the input port 4 -- vel*/
ssSetInputPortDataType(S, 3, SS_DOUBLE);
ssSetInputPortWidth(S, 3, 3);
ssSetInputPortComplexSignal(S, 3, COMPLEX_NO);
ssSetInputPortRequiredContiguous(S, 3, true);
ssSetInputPortDirectFeedThrough(S, 3, 1);
ssSetInputPortOptimOpts(S, 3, SS_REUSABLE_AND_LOCAL);

/*Set the number of output ports.*/
if (!ssSetNumOutputPorts(S, 1)) return;
ssSetOutputPortDataType(S, 0, SS_DOUBLE);
ssSetOutputPortWidth(S, 0, 3);
ssSetOutputPortComplexSignal(S, 0, COMPLEX_NO);
ssSetOutputPortOptimOpts(S, 0, SS_REUSABLE_AND_LOCAL);

/*Register reserved identifiers to avoid name conflict */
if (ssGetSimMode(S) == SS_SIMMODE_RTWGEN) {
/*Register reserved identifier for OutputFcnSpec*/
ssRegMdlInfo(S, "Delta_kane", MDL_INFO_ID_RESERVED, 0, 0,(void*)
ssGetPath(S));
}
ssSetNumSampleTimes(S, 1);
ssSetNumRWork(S, 0);
ssSetNumIWork(S, 0);
ssSetNumPWork(S, 0);
ssSetNumModes(S, 0);
ssSetNumNonsampledZCs(S, 0);

ssSetOptions(S,  SS_OPTION_USE_TLC_WITH_ACCELERATOR |
                 SS_OPTION_CAN_BE_CALLED_CONDITIONALLY |
                 SS_OPTION_EXCEPTION_FREE_CODE |
                 SS_OPTION_WORKS_WITH_CODE_REUSE |
                 SS_OPTION_SFUNCTION_INLINED_FOR_RTW |
                 SS_OPTION_DISALLOW_CONSTANT_SAMPLE_TIME);
}
/* Function: mdlInitializeSampleTimes
=========================================
 * Abstract:
 *    This function is used to specify the sample time(s) for your
 *    S-function. You must register the same number of sample times as
 *    specified in ssSetNumSampleTimes.
 */
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, INHERITED_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, FIXED_IN_MINOR_STEP_OFFSET);

#if defined(ssSetModelReferenceSampleTimeDefaultInheritance)
    ssSetModelReferenceSampleTimeDefaultInheritance(S);

#endif
}

#undef MDL_INITIALIZE_CONDITIONS   /* Change to #undef to remove
function */
#if defined(MDL_INITIALIZE_CONDITIONS)
/* Function: mdlInitializeConditions
========================================
* Abstract:
* In this function, you should initialize the continuous and discrete
* states for your S-function block.  The initial states are placed
* in the state vector, ssGetContStates(S) or ssGetRealDiscStates(S).
* You can also perform any other initialization activities that your
* S-function may require. Note, this routine will be called at the
* start of simulation and if it is present in an enabled subsystem
* configured to reset states, it will be call when the enabled
subsystem
* restarts execution to reset the states.*/

static void mdlInitializeConditions(SimStruct *S)
{
}
#endif /* MDL_INITIALIZE_CONDITIONS */
```

```
#undef MDL_START  /* Change to #undef to remove function */
#if defined(MDL_START)
  /* Function: mdlStart
=======================================================
* Abstract:
*    This function is called once at start of model execution. If you
*    have states that should be initialized once, this is the place
*    to do it.*/
static void mdlStart(SimStruct *S)
{
}
#endif /*  MDL_START */

/* Function: mdlOutputs
=======================================================
 * Abstract:
 * In this function, you compute the outputs of your S-function
 * block.*/
static void mdlOutputs(SimStruct *S, int_T tid)
{
    const real_T *Fext  = (const real_T*) ssGetInputPortSignal(S,0);
    const real_T *Tctrl = (const real_T*) ssGetInputPortSignal(S,1);
    const real_T *pos  = (const real_T*) ssGetInputPortSignal(S,2);
    const real_T *vel = (const real_T*) ssGetInputPortSignal(S,3);
    real_T       *acc = ssGetOutputPortSignal(S,0);

    Delta_kane( Fext, Tctrl, pos, vel, acc );
}

#undef MDL_UPDATE  /* Change to #undef to remove function */
#if defined(MDL_UPDATE)
  /* Function: mdlUpdate
======================================================
*Abstract:
*This function is called once for every major integration time step.
*Discrete states are typically updated here, but this function is
*useful for performing any tasks that should only take place once per
*integration step. */

static void mdlUpdate(SimStruct *S, int_T tid)
{
}
#endif /* MDL_UPDATE */
#undef MDL_DERIVATIVES  /* Change to #undef to remove function */
#if defined(MDL_DERIVATIVES)
  /* Function: mdlDerivatives
=================================================
* Abstract:
*    In this function, you compute the S-function block's derivatives.
*    The derivatives are placed in the derivative vector, ssGetdX(S).
*/
static void mdlDerivatives(SimStruct *S)
{
}
#endif /* MDL_DERIVATIVES */

/* Function: mdlTerminate
=====================================================
* Abstract:
*    In this function, you should perform any actions that are
necessary
```

```
*    at the termination of a simulation.  For example, if memory was
*    allocated in mdlStart, this is the place to free it.*/

static void mdlTerminate(SimStruct *S)
{
}

/*=============================*
 * Required S-function trailer *
 *=============================*/

#ifdef  MATLAB_MEX_FILE    /* Is this file being compiled as a MEX-
file? */
#include "simulink.c"      /* MEX-file interface mechanism */
#else
#include "cg_sfun.h"       /* Code generation registration function */
#endif
```

A.6) Delta Robot Functional Blocks

The output of the Autolev code is used to generate separate functional blocks for the implementation of simulations and experiments with the Delta Robot. For each functional block a separate C code (example is given in Section A.4) is generated and these C codes are used to build S-Function codes (example given in Section A.5). The functional blocks shown in Figure A-1 are created with the corresponding S-Function codes.

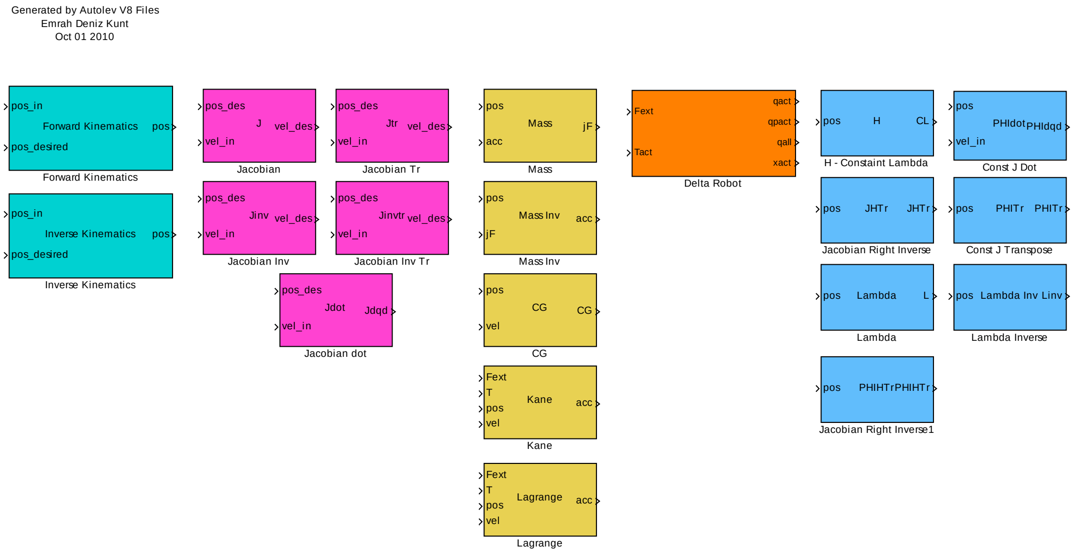
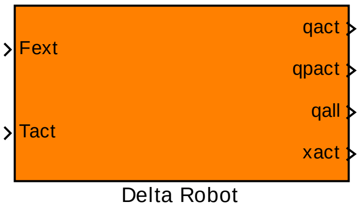


Figure A-1 – Functional Blocks



Input : External Forces in X,Y and Z, Input Torques

Output : Joint-Task Space Positions, Velocities

Figure A-2 - Functional Block for the Delta Robot

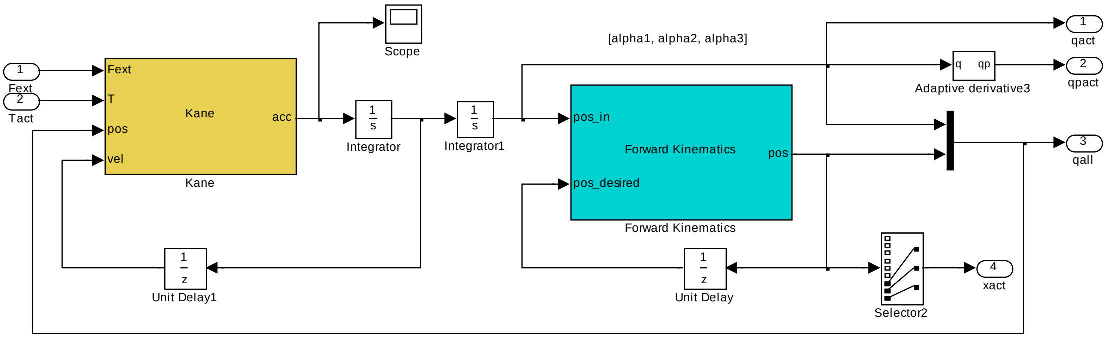
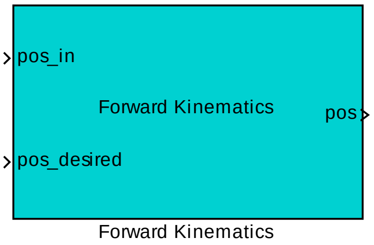


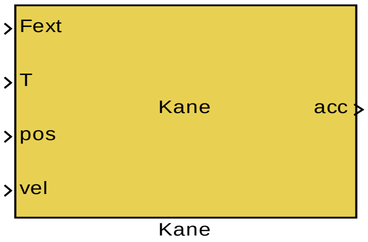
Figure A-3 – Inner Structure of the Delta Robot Functional Block



Input : Active Joint Positions, Initial Passive Joint Positions and Task Space Positions

Output : Task Space Positions

Figure A-4 - Functional Block for Forward Kinematics



Input : External Forces in X,Y and Z, Input Torques, Initial Position and Velocities

Output : Accelerations

Figure A-5- Functional Block for Dynamics (Kane)

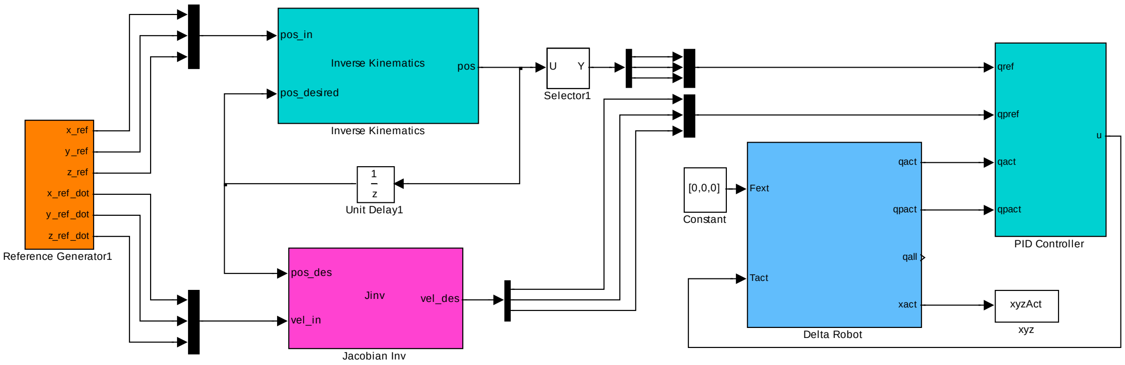


Figure A-6 - Test Simulation Using the Functional Blocks

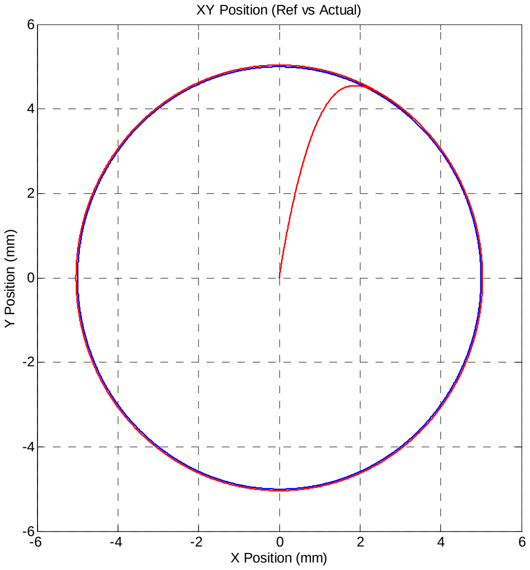


Figure A-7 - Test Simulation Results