İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

DESIGN AND IMPLEMENTATION OF AN OMNIDIRECTIONAL MOBILE ROBOT PLATFORM

M.Sc. Thesis by
Onur YAZGAN, B.Sc.

Department: Computer Engineering
Programme: Computer Engineering

JANUARY 2008

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

DESIGN AND IMPLEMENTATION OF AN OMNIDIRECTIONAL MOBILE ROBOT PLATFORM

M.Sc. Thesis by
Onur YAZGAN, B.Sc.
(504041526)

Date of submission : 24 December 2007
Date of defence examination : 29 January 2008

Supervisor (Chairman) : Prof. Dr. Ahmet Coşkun SÖNMEZ

Members of the Examining Committee : Prof. Dr. Eşref ADALI
Assist. Prof. Dr. Songül ALBAYRAK

JANUARY 2008

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

GENEL AMAÇLI BİR MOBİL ROBOT PLATFORMUN TASARIMI VE GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ

Müh. Onur YAZGAN
(504041526)

Tezin Enstitüye Verildiği Tarih : 24 Aralık 2007
Tezin Savunulduğu Tarih : 29 Ocak 2008

Tez Danışmanı : Prof. Dr. Ahmet Coşkun SÖNMEZ

Diğer Jüri Üyeleri : Prof. Dr. Eşref ADALI
Yrd. Doç. Dr. Songül ALBAYRAK

OCAK 2008

ACKNOWLEDGEMENT

I wish to express my gratitude to my supervisor, Prof. Dr. Ahmet Coşkun Sönmez who was abundantly helpful and offered invaluable support with his sincerity and belief in me.

I would also like to enunciate my love and appreciativeness to my beloved families; for their understanding, endless love and determination in supporting me.

January 2008 ONUR YAZGAN

TABLE OF CONTENTS

ABBREVIATIONS

CCD : Charge Coupled Device
DSP : Digital Signal Processor
MMACS : Million Multiply Accumulate Cycles per Second
SDRAM : Synchronous Dynamic Random Access Memory
PC : Personal Computer
MDOF : Multi Degree of Freedom
DMA : Direct Memory Access
FPGA : Field Programmable Gate Array
PHY : Physical Layer
PCB : Printed Circuit Board
CPU : Central Processing Unit
MAC : Media Access Control
MMU : Memory Management Unit
LCD : Liquid Crystal Display
TFT : Thin Film Transistor
GPIO : General Purpose Input Output
USB : Universal Serial Bus
RAM : Random Access Memory
SPI : Serial Peripheral Interface
MII : Media Independent Interface
IDE : Integrated Drive Electronics
IP : Internet Protocol
TFTP : Trivial File Transfer Protocol
DHCP : Dynamic Host Configuration Protocol
PPI : Parallel Peripheral Interface
UART : Universal Asynchronous Receiver Transmitter
TTL : Transistor-Transistor Logic
PWM : Pulse Width Modulation
RC : Radio Control
DDP : Differential Drive Platform
CAD : Computer Aided Design
DOF : Degrees of Freedom
LED : Light Emitting Diode
EMI : Electromagnetic Interference
ADC : Analogue to Digital Converter
IC : Integrated Circuit
RGB : Red Green Blue
HSI : Hue Saturation Intensity
YCbCr : Luminance Chroma Blue Chroma Red
UDP : User Datagram Protocol

LIST OF TABLES

LIST OF FIGURES

DESIGN AND IMPLEMENTATION OF A GENERAL PURPOSE OMNIDIRECTIONAL MOBILE ROBOT PLATFORM

SUMMARY

As the computation systems are taking progress with a growing rate, the facilities that were even hard to imagine once, like multi-core processors, clock frequencies far beyond thousands of megahertz and huge memory capacities are now the components of even the most ordinary computers. Resulting from this progression, processor hungry algorithms, that can generally be implemented effectively by utilizing parallel processing, like neural networks, evolutionary algorithms, colony approaches; which were grown up to become the basic building blocks, the ordinary operators of artificial intelligence applications, being implemented more easily and accurately each day. Parallel to these, fields of investigation related to robotics including computer vision, artificial intelligence, motion control, trajectory computation and obstacle avoidance have started a rapid algorithm production process.

One of the most common problems that researchers dealing with the academic studies that are taking progress in means of both quantity and quality; is the deficiency in testing their findings in a sufficient and accurate manner. The most common approach of testing a proposed method is to present the algorithm with sets of generated or collected data on a simulation platform and comparing the results to a reference algorithm presented to the same problem in the simulated environment. However these kinds of simulation based approaches are often missing many components of a real world system that can affect the overall performance. Moreover, certainly, having a chance of testing and evaluating his algorithm on a real robot is a chance that all researchers would like to have. However in practice, unless having an extraordinary opportunity, many results of obtained by researchers are stuck in the form of mathematical expressions; because having a robot platform that includes required sensors, actuators and other facilities to implement the algorithm to test is a rare situation. Moreover, even if it is assumed that a robot is available; since the robots are likely to be optimized for their goal of design, adapting an algorithm over such a robot for testing would probably require such a great amount of work that it would be infeasible.

In this thesis, design and implementation of a robot platform that is capable of satisfying the requirement of presenting an efficient development platform that enables the researchers to adapt and test any algorithm in a broad range while fulfilling the hardware requirements, was presented.

In the scope of this study, in addition to detailed expressions and mathematical analysis of the approaches used for the robot's design, the factors that are effective on the way to obtain derived results and the results obtained; two important additional applications were implemented. The first important additional work is the analysis and the application of an algorithm that calculates the position of the robot in real world coordinates, with only getting information from internal sensors by utilizing certain geometrical advantages particular to the resulting mechanical design of the mobile base, namely an algorithm to minimize the odometric errors. The second important additional work consists of the derivation, analysis and the application of a laser range measurement algorithm that that utilizes a laser pointer capable of moving around two axes and a the system camera capable of turning around three axes, which makes it possible to construct a three dimensional approximation of the environment around the robot.

Determining of the key properties that the design should have to satisfy the requirements defined above was the starting point of the work. After clarifying these constraints, designs of the mechanic, the electronic and the software components were started.

After deriving the constraints in means of robot's parameters, which assure that the required general properties, will be achieved; the design of the mechanical infrastructure was developed in stages using SolidWorks computed aided mechanical design software. At the end of each stage, the completed part of the design was modified by doing investigations, determining the components and production methods available for the completed stage, and combining the results with the experience gained through the process by experiments which resulted in a near-optimal omnidirectional mobile robot platform with significant advantages, which were analyzed in detail after completion of the implementation.

In parallel to the mechanical development, the electronic system requirements to match the global constraints were also analyzed in detail resulting in an implementation incorporating fourteen circuit boards in a sophisticated electronic structure, while satisfying the needs for generality and ease of customizing.

With the parallel developing mechanics and electronics, the basic low level software required to test the portions of the implementation were developed. After the completion of the mechanical and electrical work, higher level software to emphasis the strongest parts of the design and to provide an easy to use software base that can be used effectively for application development were implemented.

As a conclusion, after the completion of design and implementation of the system, the resulting work discussed in this text has the following properties:

- Ability to of omnidirectional movement with four-degrees-of freedom.
- Sufficient power to overcome even the most complex image processing and computer vision algorithms with a three axis camera and a digital signal processing capability of 2400MMACS.
- Enabling practically every type of algorithm to be implemented in the easiest manner with the most powerful remote monitoring and control possibilities with a control card running Linux, 128MB of volatile and 3.2GB of non-volatile memory in total and Wi-Fi network connection.

- Having a custom laser range meter, that is capable of taking range values in three axes and building a map of the environment accordingly.
- Having an accurate dead-reckoning positioning system, that gives the opportunity to provide position information without referencing any information other than the internal sensors, by utilizing properties arising from the nature of the mechanical design.

GENEL AMAÇLI BİR MOBİL ROBOT PLATFORMUN TASARIMI VE GERÇEKLENMESİ

ÖZET

Giderek artan hızla gelişen bilgisayar sistemleri; çok çekirdekli işlemciler, geçmişte imkânsız gibi görünen saat frekansları ve bellek kapasitelerini sıradan bilgisayarlarda bile bulunabilir hale getirmiştir. Bu sayede, genelde yüksek işlemci gücü gerektiren ve paralel çalışmaya daha uygun olan yapay sinir ağları, evrimsel algoritmalar, koloni yaklaşımı gibi, günümüzde birçok yapay zekâ uygulamasının temel yapıtaşı, standart operatörü haline gelmiş birçok kavram her zamankinden daha yüksek başarım ve verimle gerçeklenebilmektedir. Bu gelişmelere paralel olarak bilgisayarla görü, yapay zekâ, hareketli sistemlerin kontrolü, rota planlama ve engellerden kaçınma gibi robotikle ilgili birçok alan çok hızlı bir metot üretim sürecine girmiştir.

Sayısı ve kalitesi her geçen gün artmakta olan robotikle ilgili akademik çalışmalarda, araştırmacıların karşılaştıkları en temel sorunlardan biri, geliştirmekte oldukları metotları yeterli ve kaliteli bir biçimde test etme güçlüğüdür. Yeni bir metodun test edilmesi aşamasında genel yaklaşım, metodun gerçek sistemden tamamen izole bir bilgisayar benzetim ortamında, üretilmiş veya edinilmiş veri setlerine uygulanması ve aynı ortamda, aynı veri setlerine uygulanan bir referans metotla karşılaştırılmasıdır. Ancak bu ve benzer benzetimsel yaklaşımlar, gerçek robot sistemlerinde performansa etki edebilecek ve genelde benzetim modellerinde ön görülemeyecek donanım kaynaklı etkileri çoğunlukla yadsımaktadır. Buna ek olarak, geliştirdiği bir algoritmayı gerçek bir robot üzerinde çalışırken görmek ve gerçek dünyada performansını değerlendirmek, kuşkusuz her araştırmacının isteyeceği bir fırsattır. Ancak gerçekte, çok nadir fırsatlarla karşılaşılması dışında, bu tip araştırmalar matematiksel ifadeler halinde kalan sonuçlara gebedir, çünkü çalışır halde, gerekli duyargalara ve hareket kabiliyetine sahip bir robot platforma ulaşmak ancak bu nadir fırsatlar sonucunda mümkündür. Ayrıca, ulaşılabilir durumda bir robotun bulunduğu varsayılsa bile, robotlar genelde maliyet ve karmaşıklık optimizasyonu amacıyla, robota görev olarak düşünülmüş özel amaçlara özgü olarak tasarlanıp üretildiğinden, eldeki bir algoritmanın herhangi bir robota uyarlanarak test edilmesi, muhtemelen göze alınamayacak kadar ağır bir çalışma gerektirecektir.

Bu tez çalışmasında, yukarıda bahsedilen gerçeklerin ortaya koyduğu gereksinime yanıt vermeye yönelik olarak, üzerinde herhangi bir uygulamanın minimum çabayla çalıştırılmasına olanak verirken, çok geniş bir uygulama yelpazesine destek verebilecek kadar esnek kabiliyetlere sahip olan bir robot platform tasarlanması ve gerçeklenmesi konusu ele alınmıştır.

Bu çalışma kapsamında, robot sistemin tasarımında izlenen yolun, sonuçlara ulaşmada etkili olan faktörlerin ve ulaşılan sonucun detaylı aktarımları ve matematiksel incelenmelerine ek olarak iki önemli uygulama daha, detayları ile irdelenerek sunulmuştur. Bunlardan birincisi ulaşılan mekanik tasarıma özgü bazı geometrik avantajları kullanarak ile araç pozisyonunun sadece iç duyargalara bağlı olarak etkili biçimde hesaplanmasına olanak veren, bir başka ifade ile odometrik hataları minimize eden bir algoritmadır. İkincisi ise iki eksende hareket eden bir lazer noktalayıcı ile üç eksende hareket edebilen sistem kamerasının kullanılması ile robotun çevresinin üç boyutlu bir tanımlamasını yapabilmesine olanak veren bir lazer mesafe ölçme yöntemin elde edilmesi ve uygulanmasının detaylarını içermektedir

Çalışmaya, yukarıda tanımlanmış özelliklere sahip olmak için bir robot platformun sağlaması gereken anahtar özelliklerin belirlenmesi ve irdelenmesi ile başlanmış; bu kıstasların netleştirilmesinin ardından bunları sağlayacak mekanik-elektronik ve yazılımsal yapının tasarlanması işlemine geçilmiştir. Sistemin sağlaması gereken özelliklerin biçimsel olarak netleştirilmesinden ve matematiksel analizlerinin yapılmasının ardından bilgisayar destekli tasarım ortamında SolidWorks kullanılarak bir mekanik tasarım adım adım ilerletilmiştir. Her adımda, belirlenen parçanın pratikte elde edilebilen malzemeler ve elde bulunan işleme yöntemleriyle uygulanabilirliğinin irdelenmesi, buradan doğan geri bildirimlerin deneysel sonuçlarla birleştirilerek mekanik tasarıma yansıtılması ile sistemin tamamlanmasının ardından detaylı olarak incelenmiş olan önemli yapısal avantajlara sahip, her yöne hareket edebilen bir mobil robot mekaniği ortaya konmuştur.

Mekanik sistemin oluşumuna paralel olarak elektronik donanım gereksinimleri de incelenmiş ve sonunda toplam on dört ayrı devre kartından oluşan, oldukça gelişkin bir yapı; genellik ve kolay müdahale edilebilirlik prensiplerinin getirdiği tüm gereksinimleri karşılayacak biçimde oluşturulmuştur.

Birlikte gelişen mekanik ve elektronik tasarımların test edilmesi için temel alt seviye yazılımlar her adımda bir gereksinim olarak gerçeklenmiş ve geliştirilmiştir. Tüm sistemin belirmeye başlamasıyla birlikte, tasarımın kuvvetli özelliklerini ön plana çıkaracak ve ileride yapılacak çalışmalara kolay kullanılabilir ve etkili bir zemin hazırlayacak daha üst seviye yazılımlar tamamlanmıştır..

Bütün geliştirme sürecinin ardından varılmış olan noktada, gerçekleştirilmiş olan tasarımın ana özellikleri aşağıda verilmiştir:

- Dört serbestlik derecesi ile her yöne hareket edebilme yeteneği.
- Üç eksenli kamerası ve 2400MMACS gücünde sayısal işaret işleme devresi ile en karmaşık imge işleme ve makine görüsü algoritmalarının altından kalkabilecek kadar geniş işaret işleme yeteneği
- Linux tabanlı kontrol kartı, toplamda 128MB geçici ve 3.2GB kalıcı belleği ve Wi-Fi iletişim altyapısı ile pratikte hemen her tip algoritmanın, mümkün olan en iyi uzaktan izleme ve yönlendirme olanakları ile ve en kolay biçimde geliştirilmesine olanak verme
- Özel bir algoritma ile üç eksende mesafe ölçümü yaparak ortam haritalandırması yapabilen bir lazer mesafe ölçme sistemi
- Mekanik tasarımın doğasından kaynaklanan kendine özgü bir özelliğini kullanarak dış referanslara ihtiyaç duymadan sadece motor kodlayıcılarından faydalanan hassas konum belirleme

1. INTRODUCTION

1.1 Purpose of the Thesis

Machine vision, image and speech processing, machine learning, motion planning, perception, neural networks, swarm intelligence and many other topics, those are partially or completely involved in artificial intelligence concept, are the main fields of research in modern robotics. However, it is challenging for developers of a method, to apply and test it on a realistic platform. Evaluation of an algorithm is possible by either isolating the method or adapting it on a robot platform.

Isolating and applying the proposed method to sets of virtually generated or collected problem data on a computer is the easiest, hence the most common way of evaluation. This approach is useful when comparing the proposed method with other methods. However, it is not always possible to evaluate the performance by this kind of isolation, because other components of a robot system may always have unpredictable influences on the process; like oscillations, wind up situations, meaningless data, even reset or partial lock situations, communication errors and memory content distortions arising from hardware problems. Moreover, real-time restrictions, processing power limitations, memory capacity and many other parameters have to be concerned in practice. In summary, it is always better to apply and test a method proposal on a real world robot system if possible.

Evaluating a new method by adapting it to a working hardware is desirable in reader's point of view, but it is frequently not feasible for the developer. Having a robot, capable of processing the method, having all necessary equipment like required sensors, transducers and actuators is seldom. Moreover, it is not always easy or even possible to embed the method of interest into a robot, which is likely to be built for some other specific purpose and having proprietary hardware-software structure designed for a pre-defined use. A solution might be designing and building a hardware platform for testing the method; but concerning with mechanics, electronics and other software, just to test an algorithm is clearly infeasible.

In the scope of this thesis; a mobile robot platform, which intends to be a general development platform for academic use, was designed, built and tested. The main intention while designing the robot was keeping resources and capabilities as general as possible, thus providing an easy-to-use development and evaluation platform for a wide range of applications by supplying all necessary hardware-software infrastructure.

The robot hardware is a four-degrees-of- freedom omnidirectional mobile platform which includes:

- A color CCD camera capable of moving in three axes.
- A new laser range measurement system capable of obtaining three-dimensional readings about the environment.
- A dual core DSP board with video and audio input output capabilities, having 2400MMACS processing power.
- A main controller running Linux 2.6 at 200MHz.
- A Wi-Fi access point for data communication.
- An analog RF modulator for video and audio transmission.
- 128MB of SDRAM in total.
- 24MB of total system FLASH, additional 1GB NAND FLASH.
- 3.2GB hard disk drive.
- Power supply hardware, capable of eight hours of continuous operation.
- 2W audio output driven from controller board.

In addition, an odometry error reduction mechanism was implemented to minimize the influence of odometric inaccuracies on the user software;

If a robot system is modeled by three layers where strategic layer is responsible of generating tasks and strategies to realize these tasks, tactical layer which is responsible of generating tactics to perform a task, and an executive layer that executes the tactical plan [1]; the implementation presented in this work forms a complete executive layer with error correction, information collection and isolated low level controls on a sophisticated omnidirectional mobile platform while leaving strategic and tactical layers and large amount of processing power with various sensors, communication hardware, data storage available for developers.

The next subsection gives detailed information about the design principles and constraints, and the proposed solution of each key concept. The section clarifies how the hardware and software properties stated above are derived from these key concepts.

The second section, “System Overview”, defines the structure of the whole robot from mechanical orientation of system components to complete block description. Each electronic module in the system is briefly described during the section.

The third section concentrates on the omnidirectional moving base. The mechanic design of the base, and reasons that brought up the final design are discussed in detail. Electronic systems and mechanic systems involved in the control operation of the moving base are examined deeply, and at the final of the section approaches to reduce odometry errors are presented.

The fourth section, “Environmental Sense”, gives the details of the presented laser based range measurement system that is the base device for obtaining environment information, which makes is possible to build three dimensional layout of the environment by providing fast and accurate depth information obtained from various angles.

Additionally, the system is controlled and the system parameters are observed by a host PC, running a controller application which communicates with the robot over Wi-Fi and enables the operator to control the robot using a joystick. The PC software also receives and shows the video stream from the robot camera. The last section before the conclusion is about these remote control facilities of the robot.

The conclusion part includes brief information about the problems encountered during development phase and presents a discussion about the ways to improve the overall system performance and profitability including both the laser range measurement system, and the mechanical performances, affecting the odometric performance.

1.2 Design Principles

The mechanics, electronics and software needs of a general mobile robot development platform were determined focusing on sensory, mobility, processing power, communication, and data storage needs of most complicated algorithms in order to make the system as generalized as possible.

1.2.1 Movement Flexibility, Speed and Accuracy

The most popular conventional approach to build a mobile robot base is the three-wheel design where front wheel is steered and driven [2]. This approach is widely used for its simplicity and ease of control. The other conventional robots are usually designed as differential drive, where the robot is powered by two motors in each side, each one driving a side wheel or a track [3, 4]. These designs are easy to control, however in many cases the floor space is limited or it consists of areas where extreme maneuverability is needed [5].

Another approach to increase maneuverability, while keeping two-degrees-of-freedom is called the synchro-drive [6]. This design has three or four wheels, which are linked mechanically and steered together from a steering motor. The wheels can be steered to any direction, hence the robot can move towards any direction. However, since only the wheels are steered, the orientation of the robot body cannot be changed. Synchro-drive robots are usually cylindrical in shape because of the absence of control over the body orientation.

Ability to move towards any direction while controlling the orientation of the body is generally achieved by using special wheels, called omnidirectional wheels, that can roll sideways [7, 8, 9]. This approach still keeps the ease of control property with three-degrees-of-freedom. However, Feng [8] states that they cannot be used efficiently on rough and irregular surfaces and they are subject to inaccuracies. Moreover, omnidirectional wheels are hard to find and their load capacity is limited.

As a result, the most efficient way of obtaining true omnidirectional movement is found to be the Multi-Degree-of-Freedom (MDOF) approach. MDOF vehicles have been developed since 1920s [10], because of their great advantages in turning on a single point, following complex paths, moving sideways and moving in space limited areas.

Besides the desirable properties of MDOF designs, many problems related to them have been reported by many researchers. Reister [11] and Moravec [12] stated that many MDOF designs are difficult to control. Slippage of wheels or irregularities of movement surface cause severe odometry errors in some designs as analyzed in [1, 9, 13, 14, 15]. They also state that MDOF designs are not suitable for vehicles relying on only odometric data.

Most of robot applications rely both on absolute and relative positioning [16, 17].

Relative positioning is based on odometry, which simply computes the robot's path of movement with only the data collected from internal sensors, like number of turns of wheels and steering angle. Odometry is easy to implement and it allows inexpensive incremental encoders for the wheels [18].

Absolute positioning is usually utilizing magnetic compasses, active beacons, global positioning systems, landmarks or model matching [19]. All of these are high-cost techniques in terms of both processing complexity and money. Active beacons and landmarks require installation and maintenance, and model matching methods are highly complex and slow, and not robust [18]. Global positioning system is generally useful outdoors and accuracy of 10-30m [20] leads to errors higher than odometry most of the time.

To conclude, none of the absolute positioning methods were sufficient for general use. Moreover, system cost was considered as an important aspect. Hence, odometry was chosen for the base positioning technique, but it was obvious that some corrections had to be done, believing that as stated by Borenstein and Feng [18], if a way to improve the accuracy of odometry can be found, then the cost and complexity of mobile robot system will reduce dramatically.

A latter design proposed by Borenstein and Evans [10] has shown a great improvement of odometry accuracy with its internal odometry error correction mechanism.

The moving base design presented in this thesis grew up to be very similar to Borenstein's work, but it has no compliant linkage mechanism [21] which compensates for the momentary controller errors and makes further improvement on odometry. The compliant linkage mechanism they offer was mechanically hard to

realize and a linear encoder of extremely high resolution, which they needed was not obtainable. To compensate for the loss of odometric accuracy, a fusion technique between the odometric calculations and the data obtained from the laser range measurement system can be utilized, like in the works of Cox [22], Byrne [20], and Chenavier and Crowley [16], who proposed methods of cross-checking odometric results with referenced position measurements.

1.2.2 Sufficient Internal and External Sensing

There are various types of external sensors for mobile robots like ultrasonic proximity sensors, infrared proximity sensors, tactile sensors, time of flight laser rangers, phase shift and Doppler shift rangers. While the common purpose and basic usage area of all these sensors are obstacle avoidance; each of these sensors provides its own advantages and disadvantages.

Ultrasonic sensors are easy to implement and cheap. Their range of reading is suitable for indoor robot applications and their accuracy is good at ideal conditions. However, they suffer from poor directionality and reflections. This makes it hard to develop a map based trajectory control system using this kind of sensors.

Directionality problem of ultrasonic sensors does not exist with infrared proximity detectors; however the range of about 30cm at maximum is less than required minimum to obtain meaningful data. The same problem exists with tactile sensors.

Laser range meters are excellent in both accuracy and directionality. There are commercial laser range meters for industrial applications, which can measure distances up to 1500m with sub-centimeter resolutions. However, both time of flight laser rangers and phase shift laser rangers utilize special optics, avalanche photodiodes, picoseconds timers or two stage mixers and more complicated components, which make them bulky and costly.

The robot should clearly include a camera and an image processing subsystem for the goal of being a general application development platform. By utilizing this camera and spending a little portion of the image processing power, a method for measuring the distance of a laser spot in a scene has been realized. Moreover, to make it possible to build a three-dimensional overview of the environment, the laser beam was enabled to be directed to any point in the field of view of the camera.

It is not needed to have color images to detect a laser spot in a scene. Moreover, monochrome CCD cameras are cheaper and it is possible to use a very high quality monochrome camera with low-light vision, instead of an ordinary color CCD camera. However, by having the ability to measure the distance of any point in the field of view of the camera, one essential problem in machine vision, which is also foreseen to be one of the most frequent usage reasons of the platform, can be solved more effectively: background segmentation. Although processing of high resolution gray-scale images often provides sufficient results [23], it needs more complicated algorithms, hence more processing power; and there are many methods in literature for background segmentation by color segmentation. Color is known to be robust and effective visual information for distinguishing an object from others [24]. However, it is known that color segmentation suffers from shadows, color variations and irregular illumination. Gordon [25] states that neither range nor color is sufficient by itself for background segmentation; because depth measurements are not available at each pixel and objects may be indistinguishable when close to background and color suffers from the problems mentioned above. He proposes a method for using both range and color information to remove the background from the scene and segmentation of objects, and he achieved impressive results. Hence, by assuming any effective segmentation algorithm should need color information, a color CCD camera was chosen in expense of quality decay.

Most of the internal sensors are invisible to the topmost layers, where the user will do his development. These sensors are the incremental optical encoders at four drive motors, multi-turn potentiometers to measure to angles about the base and internal potentiometers of five servo motors used for camera and laser movements. The data collected from these sensors are normally processed by the executive layer to correct odometry errors, move to a target point and generate depth measurements. The user does not need to concern with these variables unless a change in executive layer is needed.

There are only three types of information obtained from internal sensors and passed directly to upper layers, those are the voltage of the batteries, instantaneous current and state of being operated from batteries or AC power.

1.2.3 Sufficient and Flexible Processing Power

Algorithms for computer vision, namely image processing algorithms are generally the most processing power and memory consuming tasks among all. Real time processing of video stream obtained from system camera at a resolution of 720 by 576 pixels with 32 bits per pixel needs processing of 41472000 bytes of pixel data in a second at 25 frames per second. The objective of being general, forces the design to include a huge processing power; however this makes a contradiction with two of the design's other objectives. Those are being power efficient and ease of software development.

Microprocessors with such amount of processing power are often complicated semiconductors with serious power consumption. Moreover, since these processors are not designed directly for image processing, they generally do not have a pixel port, enabling Direct Memory Access (DMA) controlled data transmissions and receptions from and to pixel devices. This brings up the concept of frame grabber subsystems, which in turn makes the system even more complicated and power consuming.

Using dedicated image processing semiconductors, or arrays of FPGA built for image processing is a very power efficient way of adding image processing capability to an existing system, but only when the operation to be done on the image is pre-defined and the FPGA code is optimized or the custom IC is selected accordingly, which is clearly not a generalization approach.

Using Digital Signal Processors (DSPs) is the most convenient way to deal with low-power, high density work loads of a wide scope. These are often performance optimized devices with a broad range of input output ports including pixel ports. However, these devices do not have memory management units; hence it is not possible to set up a standard Linux distribution on them. Custom Linux distributions like ucLinux or other operating systems like VxWorks are available for many DSP families; but because these are not standard operating systems, developing applications, remote access and using third party software would be hard, distorting the aim of easy development.

As a solution to contradictions above, a two stage structure was needed. One of them, the controller board, would include a standard low-power microprocessor with Linux

2.6 on it, enabling the user to develop applications on PC and directly test them on platform even by mounting a PC folder and executing the program by connecting to the robot via Telnet. The second block would be a DSP board, which would be selected as powerful as possible, and which would have required circuitry to interface with audio and video input-output devices. It is always possible to upload a program to the DSP board via the controller board using Telnet.

1.2.4 Communication Facilities

In the development phase of an algorithm, capturing parameters, input-outputs and debug messages in real time is a very useful ability. The robot has to be able to provide this information to a remote location in real time.

In addition, for the ease of software development, it is preferable to be able to connect to the console of the controller board via Telnet, and to be able to mount shared network folders to get rid of having to upload whole ramdisk in any minor change during development. In conclusion, it was clear that the robot needed a wireless Ethernet connection.

Most of the image and speech processing algorithms form complicated software structures when realized. Developing these methods in embedded platform can be sometimes time consuming. It may be logical to develop the application or a part of it on PC; then cross-compile and embed it on to the robot. For this reason, it is assumed that real time video and audio streaming is needed. However, transferring video stream over the Ethernet connection requires encoding of the frames and consumes a lot of processor power both at the DSP side when compressing frames, and the controller board when forwarding the data to the Ethernet. In addition, these processes take a lot of time and cause a significant lag in the reception of the video stream. To avoid these disadvantages and handle video and audio transmission, a simple radio frequency modulator would be used, and thus a huge amount of information would be transferred to a host PC which is equipped with a radio frequency demodulator and a frame grabber, without using any processor power or communication bandwidth on the mobile robot.

2. SYSTEM OVERVIEW

This section gives a brief overview of the presented system. The main flow of this section does not include information about the moving base, which is examined in detail in the next section.

2.1 Technical Specifications

Table 2.1.1 below contains the technical specifications of the system and Figure 2.1.1; Figure 2.1.2 and Figure 2.1.3 are the photographs of the actual prototype taken from different perspectives.

Table 2.1.1 System Specifications

GENERAL SPECIFICATIONS	
Weight	10.4kg
Dimensions	275mm x 550mm x 200mm
Maximum Speed	0.29m/s
Battery Voltage	12V
Battery Capacity	12Ah
Input Voltage	85-220VAC
Nominal Power	15.6W
Total Motor Power	31.7W
Power Regulator Outputs	5.0V @ 3A, 3.3V @ 3A, 14.1V @ 3A
Wi-Fi Module Standard	802.11b/g
LCD	128x64 STN Graphic LCD Module
MAIN CONTROLLER SPECIFICATIONS	
Processor Type	ARM920T (Cirrus Logic EP9307)
Processor Speed	200MHz
System RAM	64MB
System FLASH	16MB
Hard Disk Capacity	3.2GB
Auxiliary FLASH	1GB NAND
DSP SYSTEM SPECIFICATIONS	
DSP Type	Analog Devices BlackFin BF561-600
DSP Clock Frequency	2x600MHz Dual Core
DSP Processing Power	2400MMACS total
DSP RAM	64MB
DSP FLASH	8MB
Input Video Port	3 Channels Composite Video, RGB or YUV

Output Pixel Port	3 Channels Composite Video, RGB or YUV
Input Audio Port	4 Channels 96kHz Sampling Rate
Output Audio Port	6 Channels 96kHz Sampling Rate
CAMERA SYSTEM SPECIFICATIONS	
Horizontal Resolution	720 pixels
Vertical Resolution	576 pixels
Camera Type	1/3'' Sharp Color CCD with 6mm CCTV Lens
Pitch Angle	30˚
Yaw Angle	180˚
Roll Angle	180˚
LASER SYSTEM SPECIFICATIONS	
Laser Type	Class III Green Laser
Laser Power	50mW
Pitch Angle	130˚
Yaw Angle	180˚

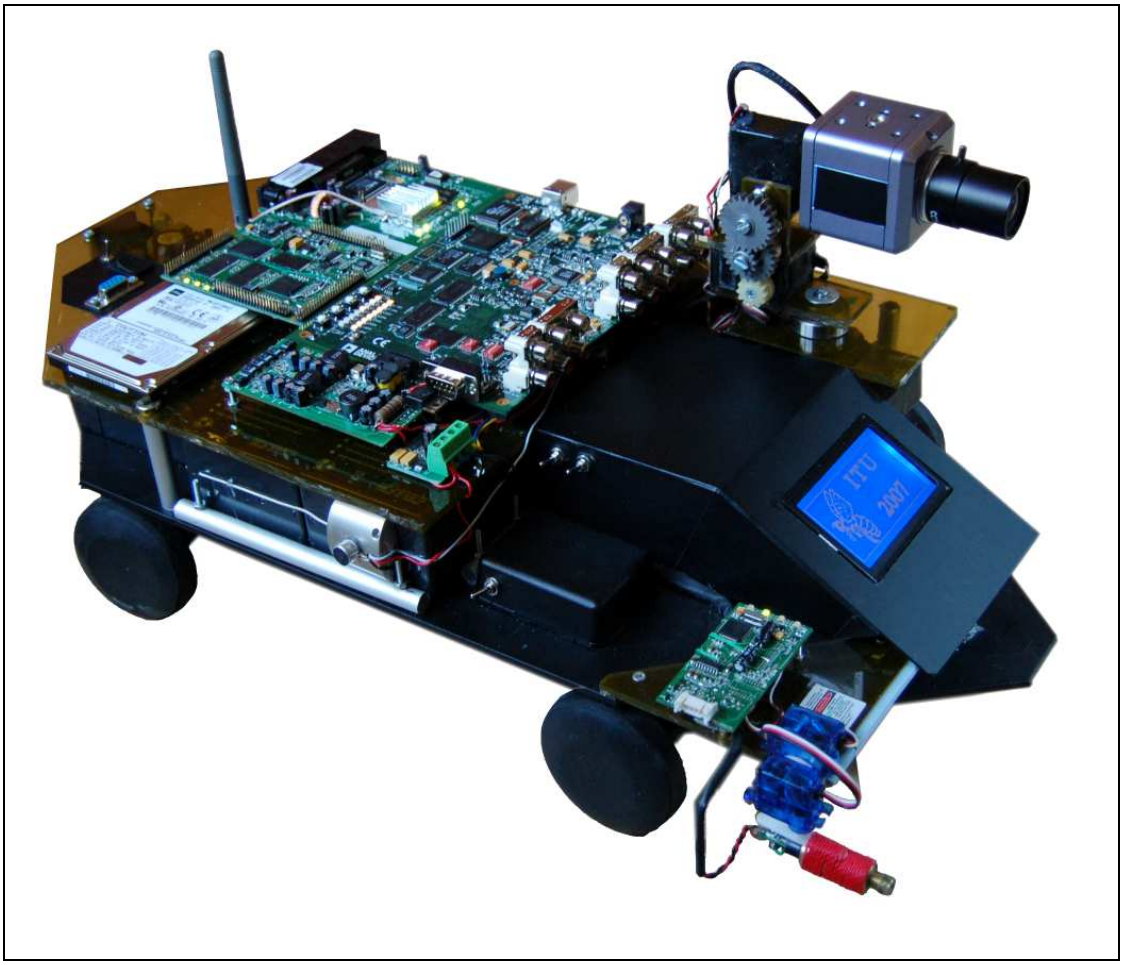

Figure 2.1.1 Upper Right Perspective of the Robot

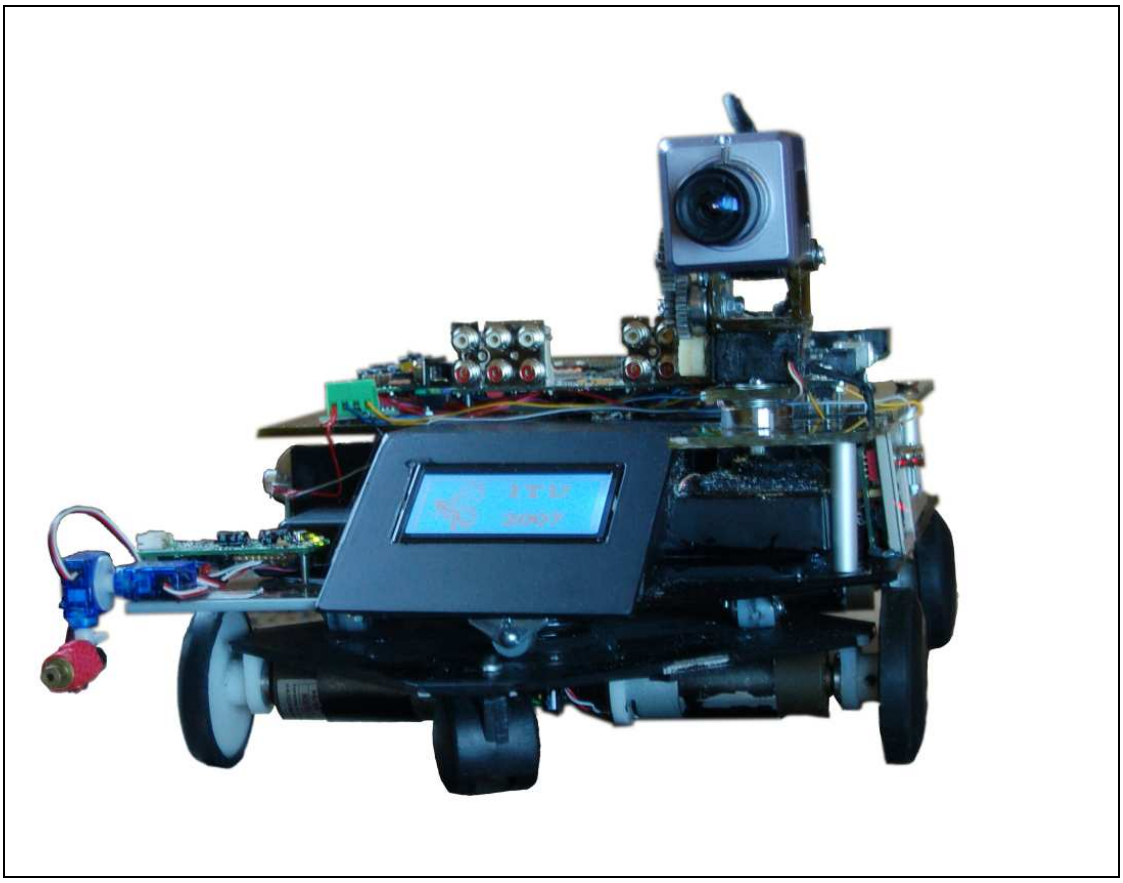

Figure 2.1.2 Frontal View of the Robot

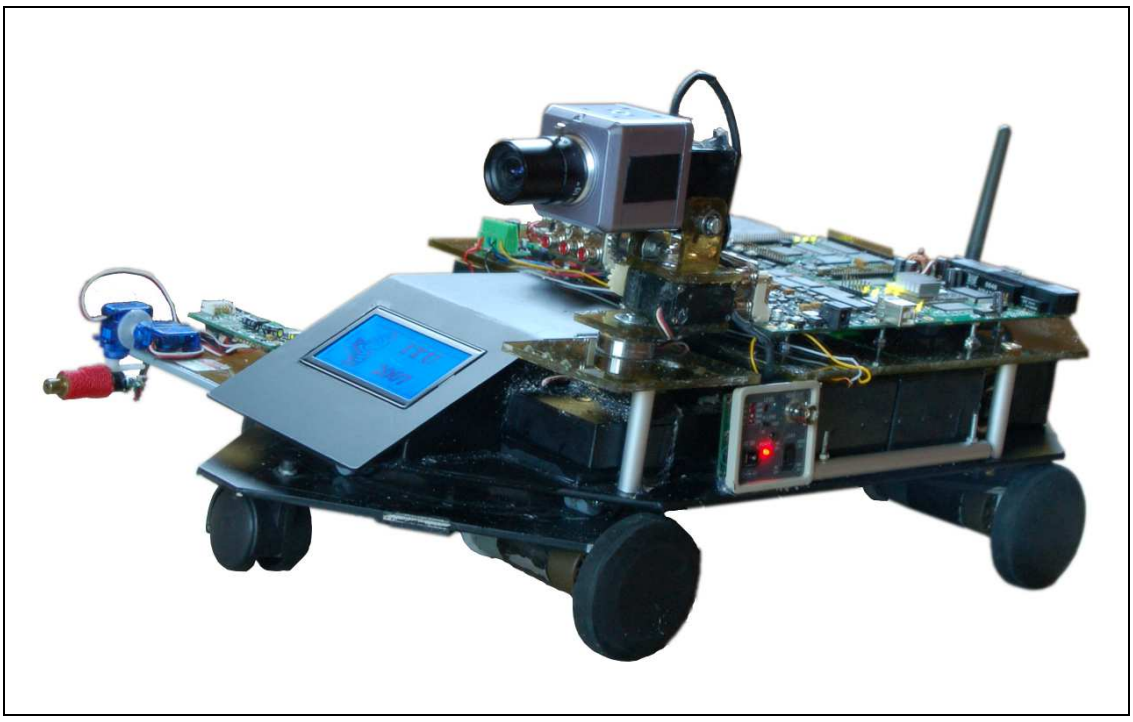

Figure 2.1.3 Upper Left Perspective of the Robot

2.2 System Electronics

The system electronics consist of fourteen separate circuit boards. The main aim in separating the electronic cards was to separate tasks to independent controllers and isolate management of specific issues from each other for faster debugging of both the electronics and the firmware. Two of these circuits, the DSP board and the Wi-Fi access point were not designed; instead they were bought and adapted for the reasons that will be presented later in this section. The most important ones among these circuits will be briefly described one by one in this section, and at last the rest of the circuits will be described under one heading.

At the end of this section, a block diagram to clarify the connections between these cards and the type of connections between them, excluding the power subsystem and power connections; and a figure showing the placement of major cards are provided.

2.2.1 Main Controller Board

Main controller board is actually a mini-computer board with RAM and FLASH memories, clock oscillators, real time clock, audio codec, Ethernet PHY and extra NAND FLASH on board. It was designed in the form of a module and every signal from the CPU is brought to edge connectors for future expansion.

The main controller board is a specially designed, six-layer PCB. The processor used is the Cirrus Logic EP9307. EP9307 is an ARM920T core microprocessor, running at 200MHz. It is equipped with a special floating point engine, Linux and Windows CE enabled MMU, LCD raster engine, Ethernet MAC and many other peripherals [26]. System RAM consists of two Samsung K4S561632 SDRAM chips on board, each one is 32MB arranged as 4M words at 4 banks. The system boots from an Intel J3 StrataFLASH chip, which is 16MB in capacity. This capacity is enough for basic Linux setup and initial ramdisk. Once the system boots, it can run applications from USB, NAND FLASH, hard disk or an Ethernet path.

The expansion connectors have the Serial Peripheral Interface (SPI) bus, data bus, address bus, control signals, the Medial Independent Interface (MII), LCD pixel port capable of directly driving digital TFT LCD modules, audio inputs and outputs, three USB ports, 19 GPIO ports and three TTL serial ports.

A simplified block diagram of the main controller board is shown below at Figure 2.2.1.1.

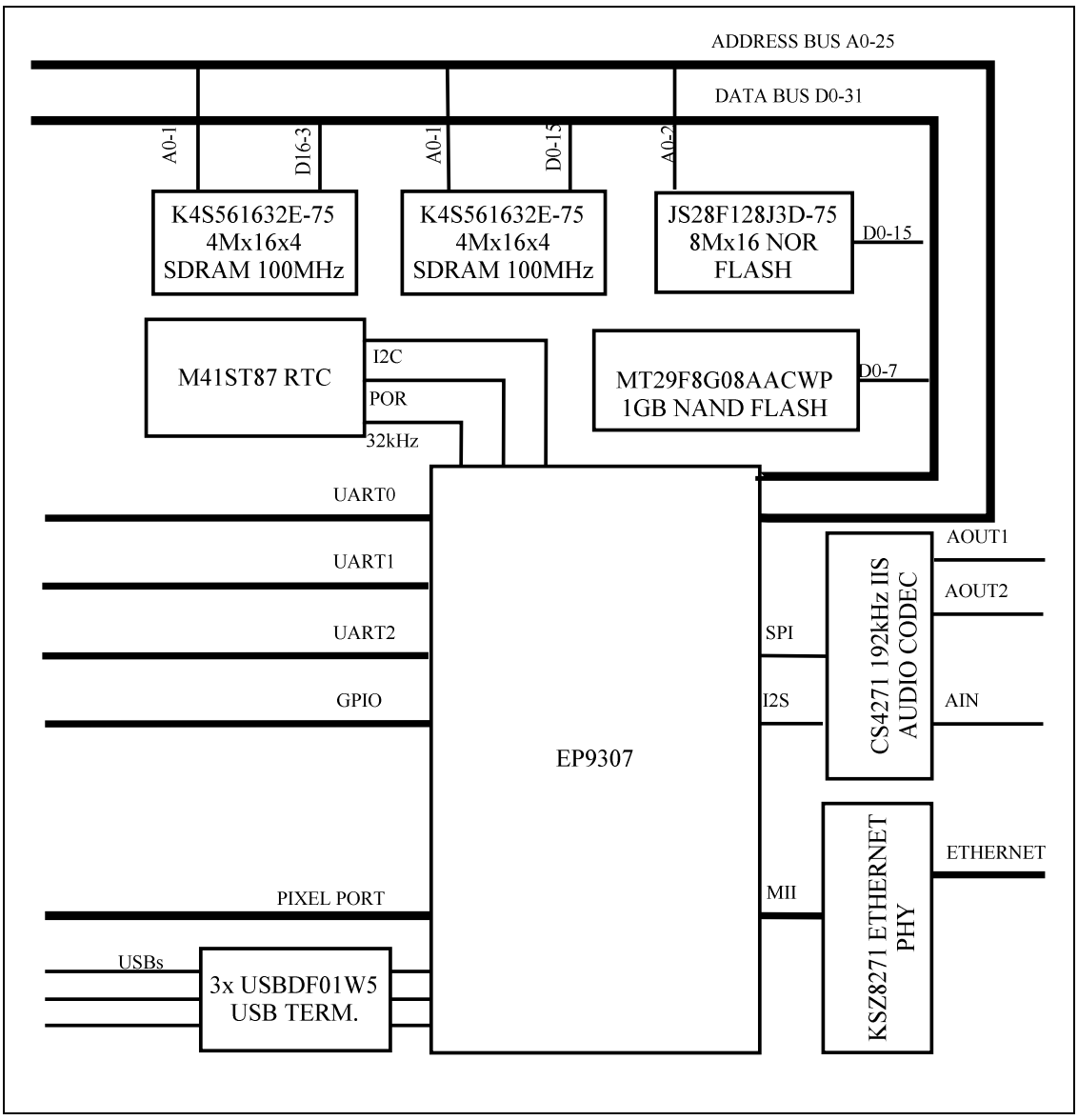

Figure 2.2.1.1 Block Diagram of Main Controller Board

The pixel port, second channel of the audio output port, audio input port, GPIO ports, address bus, data bus, control signals, second USB port and third USB port are not used on the system, they are left on the pin headers around the controller board for future use.

First serial port of EP9307 is connected to a DB9F connector on chassis via a RS232 transceiver circuit and it is used for console. The second serial port is connected to the serial port of DSP board, used for internal communication with the DSPs. The third serial port is multiplexed between the “Camera Position Controller and LCD

Bridge Card" and the "Laser Position Controller and Base Control" card, those will be defined later in this section.

The USB port is connected to a USB to IDE bridge circuit and provides connection to the IDE hard disk drive present.

The first differential audio output channel is used to drive the system speaker through a differential preamplifier and a 2W mono bridge amplifier.

The Ethernet port is connected to one of the Ethernet ports of the Ethernet switch of Wi-Fi access point module via an Ethernet pulse transformer circuit.

The main controller board requires only a stable 3.3V source to operate. The flash is loaded with a distribution of Redboot using a serial downloader program provided by Cirrus Logic. Once the Redboot bootloader is up, it is used to download ramdisk and kernel images from a TFTP server and these images are burnt into the FLASH.

Once the kernel and ramdisk images are present in the FLASH, with the initialization of the boot script, the board wakes up, boots Linux, loads and unzips ramdisk, brings up the Ethernet interface and tries to get an IP address via DHCP. Once the address is taken the system parameter reporter, the remote controller and the path manager applications, specifically written for remote control, test and benchmarking of the robot platform a start.

More information about the processor and the Linux distribution for the processor can be found in reference [26] and the at website of the Cirrus Logic.

2.2.2 Digital Signal Processor Board

Analog Devices's BlackFin DSP family consists of high performance digital signal processors, specifically designed for embedded applications. Moreover, since Texas Instruments still holds a significant part of the DSP market, Analog Devices tries to increase his portion by providing development boards, development software, technical assistance and samples easily at a very low price. To use these advantages, the most powerful part from the BlackFin family was chosen as the DSP.

The BF561 DSP is a dual core DSP which has two BlackFin DSP cores running independently 600MHz. Thus, it provides two independent powerful processors that

can either be used for different processing jobs or phases of a single job. This brings the most powerful and general purpose processing facilities on board.

As a result of the competition stated above, Analog Devices provides the development kit of BF561, which includes 64MB SDRAM, 8MB Flash, a video encoder, a video decoder and an audio codec; providing three channels of video input, three channels of video output, four channels of audio input and six channels of audio output [27], at a price that is about half the price of producing the same circuit as a prototype. Although the circuit schematics, firmware, even the Gerber files of the PCB for production is given, and the samples of the integrated circuits used on board are easy to obtain; the production of a six layer PCB is still more expensive than buying a ready-to-use development board from Analog Devices. Hence, this circuit is not re-designed; it is used in its original form adapted to the robot's structure.

Figure 2.2.2.1 below shows the general structure of the DSP board [27].

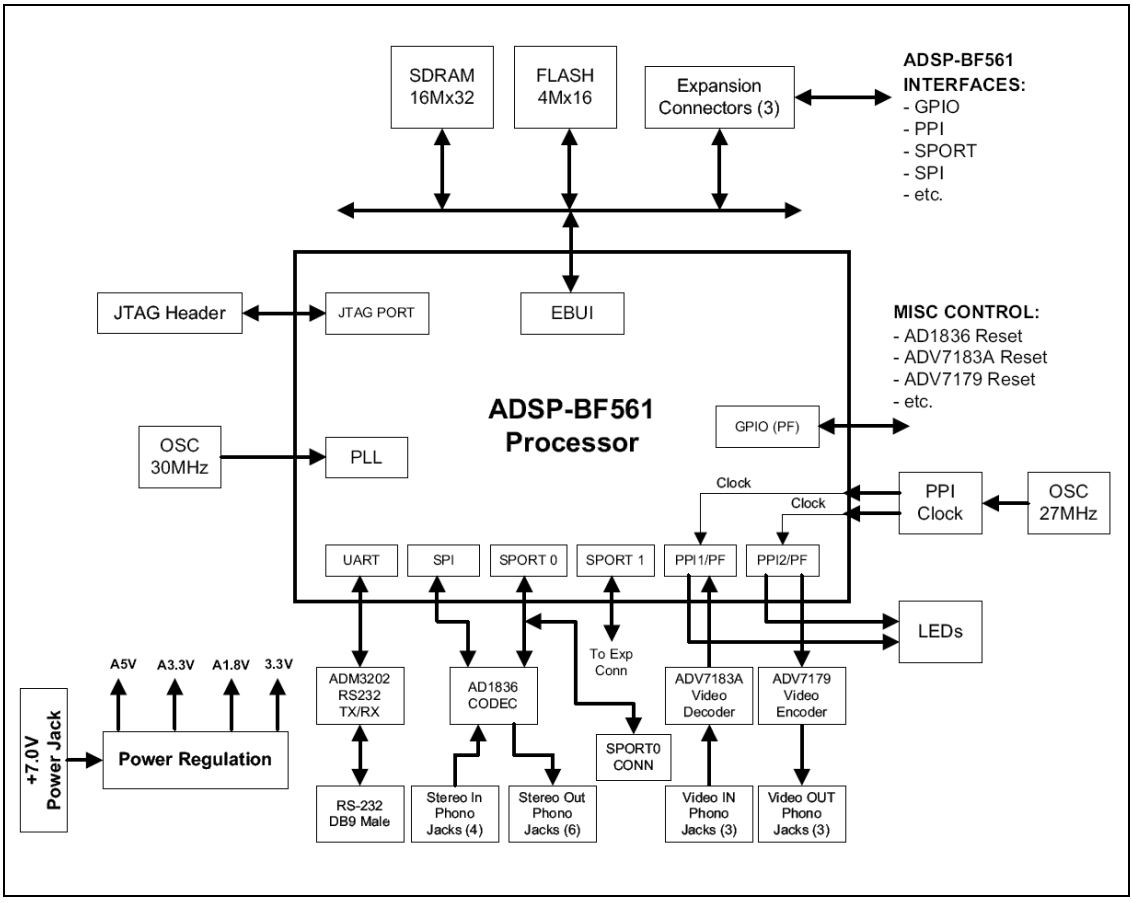

Figure 2.2.2.1 General Structure of the DSP Board

One of the video input ports are connected to the system camera. The composite video signal from camera is decoded by the ADV7183 video decoder [28] chip on board and the data stream is taken in prom PPI1 in ITU-R 656 format [29] of the DSP by a DMA channel. After processing the frame, another DMA channel puts either the process result or the input image data in ITU-R 656 format on the PPI0 of the DSP, which is connected to ADV7179 video encoder [30], which provides video output to the RF video modulator and transmitter circuit.

The serial port of the board is brought down to TTL levels by an external RS232 transceiver circuit and connected to UART1 of the main controller board for internal communications.

2.2.3 Camera Position Control and LCD Bridge Card

The camera of the system is capable of moving in three axes as stated above. Figure 2.2.3.1 below shows the structure of the camera system.

Figure 2.2.3.1 The Camera System

These movements are achieved by commercial RC servo motors used in radio controlled model aero planes and helicopters. These RC servos are also used for building many degrees-of-freedom robots, because of their ease of interfacing, extreme torque to weight ratio and small housings [31]. Figure 2.2.3.2 shows the general structure of a RC servo.

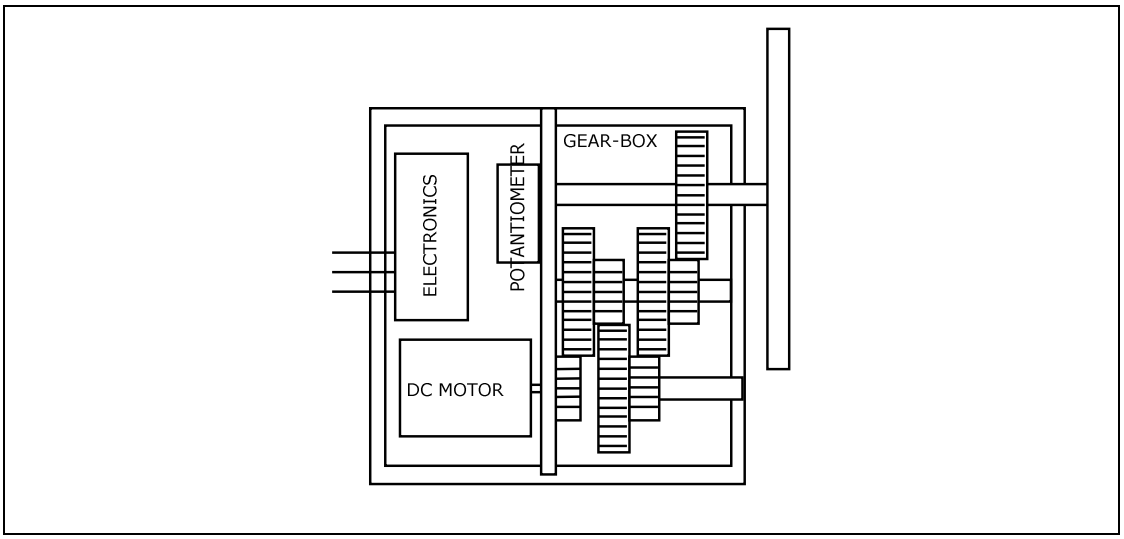

Figure 2.2.3.2 Structure of a RC servo

The servo has a three-wire interface. Two of these wires are used for power supply, namely +5V and ground connections. The third pin is used to control the position of the servo shaft. The desired position is coded by pulse width modulation. The period of the pulse train is 20ms. The duty cycle of the pulses is changed, resulting in a position change on the output shaft. A pulse width of 1.5ms which makes a %7.5 duty cycle moves the servo shaft to the center position which is referred as 0 degrees. Increasing or decreasing the pulse width by 10μs results in about one degree of rotation on the servo shaft. Hence, the servo goes to its lower limiting position of -45˚ if the pulse width is 0.6ms and to its upper limiting position of 45˚ if the pulse width is 2.4ms. The only task to be performed to control a RC servo is to change the duty cycle of a 50Hz PWM signal accordingly. The internal electronics of the servo does the rest of the work. The most of the cheapest RC servos involve an error amplifier that compares the average of the pulse train voltage to the voltage divided by the shaft potentiometer. Although these kinds of servos are sufficient for most of the modeling purposes, they tend to have oscillations at heavy loads. A second kind of more expensive servo motors with digital position control algorithms are available, and the camera movement system utilizes this kind of servo motors.

The camera control and LCD bridge card simply consists of a popular ARM7 microcontroller from NXP Semiconductors, the LPC2138, the reset circuit and power circuit. The three servo motors, each one moving the camera on one axis are controlled by three channels of PWM generators present in LPC2138. The LPC2138 is connected to third serial port of the main controller and it accepts special

commands telling to move each motor to a specified angle. The microcontroller simply calculates the duty cycle of the relevant PWM channel and the register values to achieve the calculated duty cycle; then it applies the changes resulting in a mechanical position change on the relevant servo.

In addition, a 128x64 graphics LCD module is present on the front side of the robot platform, to use in general. This LCD module is connected to one of the ports of the LPC microcontroller. The module is a HD44780 based standard graphic LCD module which is driven with parallel bus. The microcontroller translates the data sent by the main controller to be displayed on the screen to the LCD module controller.

2.2.4 Laser Position Controller and Base Control Card

The laser diode present on the system is capable of moving in yaw and pitch directions by two RC servos. Theses servos are similar to ones used for camera movement, but they are smaller and have less torque, which is not needed to move the laser diode. Controlling of these servos is done exactly as in the camera control system. Only an additional command to turn the laser on and off is added.

Figure 2.2.4.1 The Laser Diode and Driver Servo Motors

The card utilizes another LPC2138 ARM7 microcontroller from NXP. This card is a critical component of the moving base because it measures the angles relevant to the base joints, reads the speed and distance information from the motor control cards and applies appropriate speed change commands to the motor controllers to achieve a given set of angles calculated internally to compensate with odometry errors. Details of this system and the whole process are clearly defined in Section 3. All the

odometry error reduction and moving base control algorithms are performed by this card. It accepts new real world coordinates from the main controller, or simple commands like turn, go forward, reverse, sideway and stop.

The ARM7 microcontroller of the base control card is connected to the second serial port of the main controller through one of its serial ports. The other serial port is connected to two motor control cards, each of which is capable of controlling two drive motors. The potentiometers that measure the joint angles, those will be defined in Section3, are connected to two analog to digital converter channels of the ARM7. The laser servo motors are driven by two PWM channels, while the laser is turned on and off by a relay triggered by a GPIO port of the microcontroller.

The working principles of the base control card and the motor control cards are inconvenient to explain without prior knowledge of the moving base design. Hence the both hardware and software details of these cards are discussed in detail in Section 3.

2.2.5 Power System

Power system consists of six lead-acid batteries, an AC switch mode power supply and a power card consisting of three switch mode buck converters.

The power pack of the system is formed by six lead acid batteries; each one is providing 6V with 4Ah of capacity. The batteries are parallel in groups of three, and two groups are serially connected forming a 12V, 12Ah power pack.

The AC adapter is used for supplying the DC voltage that enables charging of batteries. The power supply accepts 85-230VAC input and delivers 24V at 1A. This voltage is down converted to 14.1V by the first buck converter on the power card, and this voltage is fed to the battery pack over a constant current regulator for charging. It is also fed to the other two step-down converters that provide 5V and 3.3V powers required for robot circuitry. Hence it is possible to continue working with the robot when the system is in charge, or it is possible to disconnect battery pack completely by battery switch and run on AC power only to protect batteries.

There are also local voltage regulators on each card of the system to be able to use relatively high voltage on power supplies and hence keeping the regulators safe out of saturation, while reducing the possibility of spurious reset situations that can be

caused by the EMI radiated through the motors involved in the system. Moreover, by instantaneous powering up of the drive motors, the power supplies are brought down a couple of volts before they can compensate for the new power consumption, which in turn might cause low voltage oriented reset situations if the cards were powered entirely from 3.3V or 5V supplies.

Figure 2.2.5.1 below shows a schematic layout of the power system of the robot.

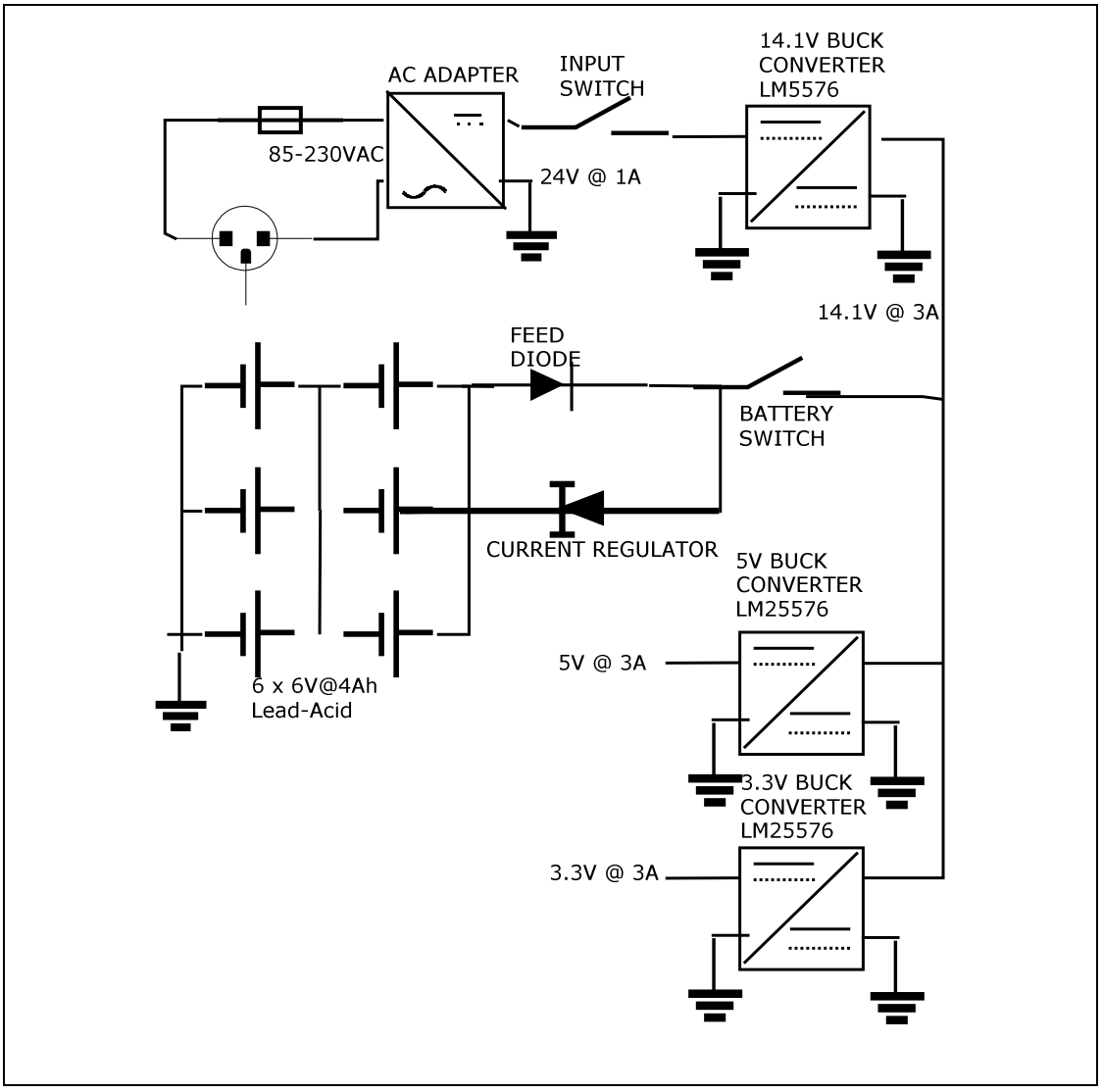

Figure 2.2.5.1 Schematic Layout of the Power System

2.2.6 Other Circuits

There are seven circuits in additions to the ones described above. Most of these circuits realize an interfacing function between parts of the robot. There are two exceptions; the Wi-Fi module and the RF modulator.

The Wi-Fi module is a commercial access point by Philips that has been adapted to work with the robot hardware. The antenna has been replaced, and the first stage if the power supply circuit was by-passed to match the available supplies.

The RF modulator is functioning as a transmitter, transmitting video and audio data in analog format by amplitude modulating the composite signals to radiate and reach a demodulator present beside the host.

There are two RS232 transceiver circuits, based on MAX232E, one of which is used for the console port and the other is used to interface the DSP board's serial port to the TTL serial port of the main controller.

There is a USB-IDE bridge, which is one of the commercially available cards those are found in external hard disk boxes. It interfaces the hard disk to the USB port of the main control board, and Linux detects the hard disk as a mass storage device attached to USB.

There is an audio analog front end circuit containing a differential preamplifier and a bridge amplifier for the speaker. The first version of this card was designed using a single ended preamplifier, and only one of the differential outputs of the audio codec on the main controller board was used for speaker, however the PWM frequency used to drive servos, the noise that system drive motors induce on power and ground lines and the EMI from the digital lines, that are travelling a distance over the amplifier inside the robot body resulted a highly distorted, noisy operation. Hence, the card was updated using a differential approach and reducing the noise levels down to a plausible limit.

The last circuit is an analog front end of the Ethernet PHY on the main controller board. It consists of termination networks and a pulse transformer to match one of the Wi-Fi access point's switch ports to the Ethernet PHY.

2.3 Internal Connections

Although the connections from and to each card were stated previously in this chapter, it is hard to visualize and understand the general picture by partial information. The internal connections of the robot are such complicated that the total length of cables used for internal connections is more than fifty meters. The block

diagram below at Figure 2.3.1 summarizes the connections and their electrical types between the components of the system. Figure 2.3.2 is an annotated photograph of the top view of the robot, which clarifies the placement of major electronic subsystems in order to visualize system structure easily. The "Camera Position Control and LCD Bridge Card" is inside the black box that holds LCD on the front panel, the motor controllers are underneath the robot and all other cards are below the topmost platform where main system cards lay.

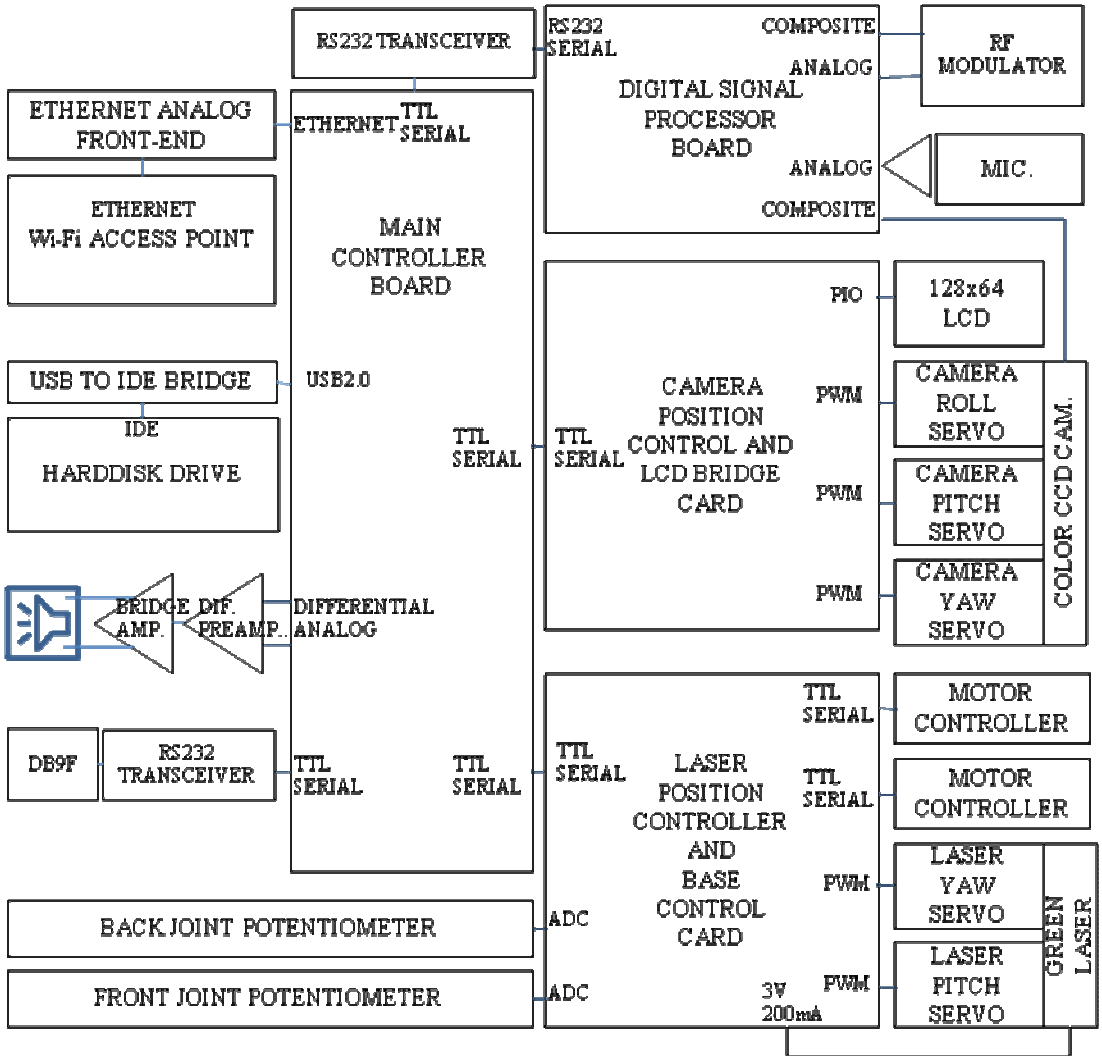

Figure 2.3.1 Interconnection Block Diagram

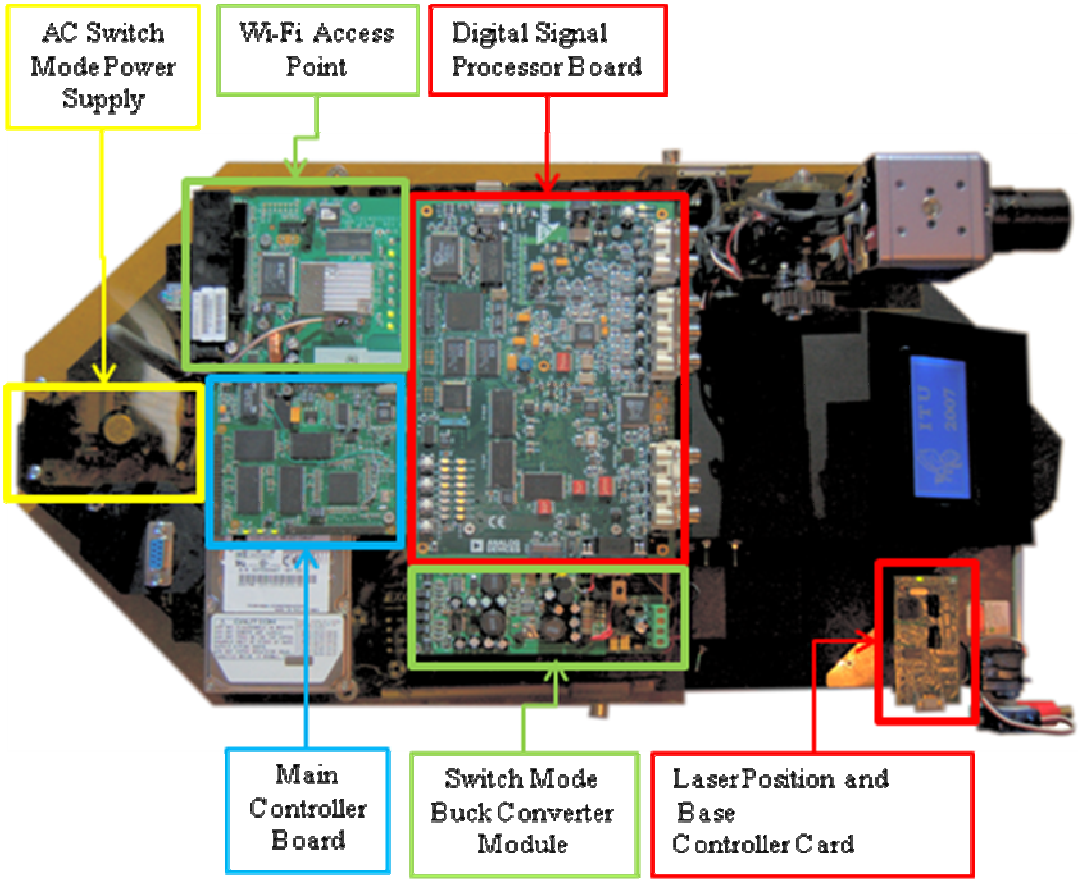

Figure 2.3.2 Placements of some Major System Component Cards

3. OMNIDIRECTIONAL MOBILE ROBOT BASE

Section 3, includes detailed information about the design and implementation of the omnidirectional mobile base platform of the robot.

Section 3.1 below, describes the approach and occurrences that resulted in the current mechanic design. After stating the process that brought up the final mechanics; Subsection 3.2 gives detailed information about the implementation and analysis of the proposed mechanical system.

The third subsection analyzes the software and hardware structure of the electronic circuits, called “Motor Controllers”. The placement and connections influencing the mechanical design is covered in second subsection, whereas the third subsection concentrates on the responsibilities of these modules and the algorithms that were built accordingly.

Section 3.4 focuses on the low level control of angles of the two “Differential Drive Platforms (DDPs)”. Detailed information about the function and design of the DDPs would be presented in Section 3.1 and Section 3.2 before dealing with angle control of them.

The last subsection, Section 3.5, in starts with the sources and reasons of odometry errors, and concludes with methods presented to reduce the odometry errors.

3.1 Creation of the Mechanical Design

The key points, the objectives expected to be satisfied by the mechanic design of the mobile robot base were covered by Section 1.2. These objectives can be summarized as being omnidirectional and accurate with reducible odometry errors.

Designing, or finding an appropriate design for clearly stated objectives is not a difficult issue, however dealing with mechanics, especially with a limited budged and sources, forces the designer to adopt an induction based approach, instead of a reduction from the objectives to the system components.

It not always possible to find suitable components like motors, encoders, gearboxes and fixing mechanisms for a design constructed to meet given objectives without having a wide range of components available off the shelf. Moreover, this kind of approach usually requires custom made components that should be produced specially by professionals using lathe and borer machines, which is quite time consuming and costly.

Basing on the reasons explained above, before starting the mechanical design, a search for mechanical components, and available mechanical processes was done. Since the main determining part of the mobile base is the encoder-motor-gearbox triplet, the search focused on finding these kinds of motors that are available at affordable prices, ending up with the complete system shown on Figure 3.1.1.

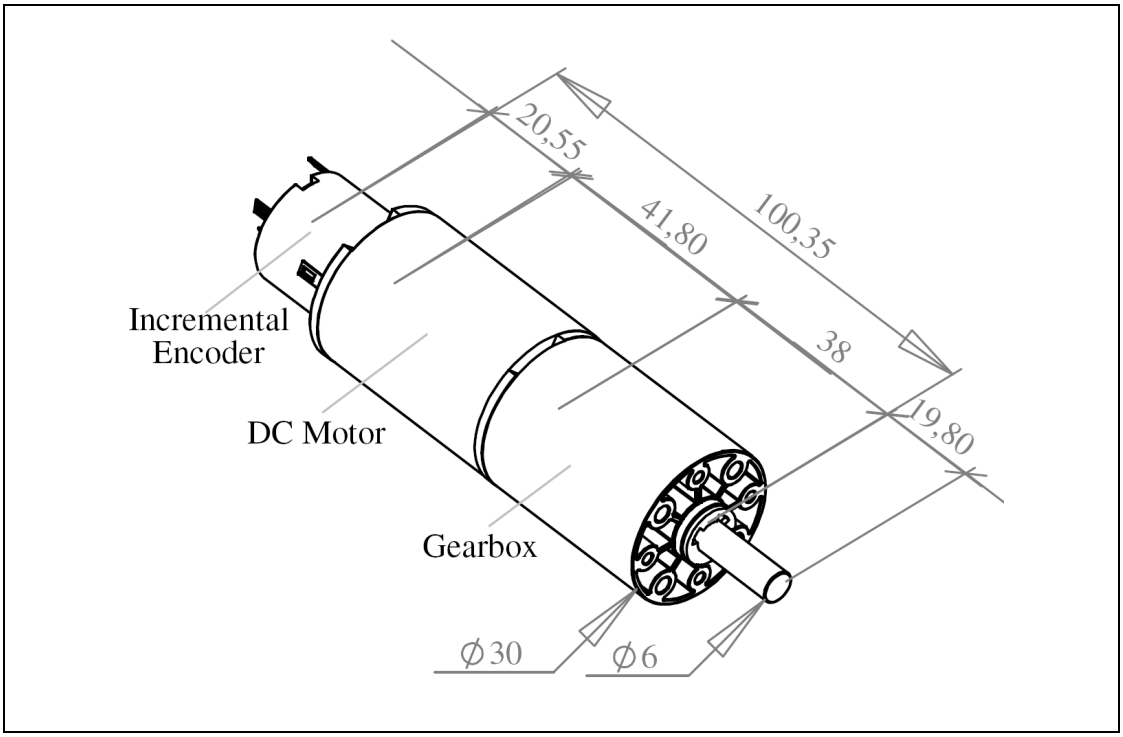

Figure 3.1.1 Drive Motor with Incremental Encoder and Gearbox

A CAD design phase was started using SolidWorks after modeling the available motor shown on the figure above. The main constraints in that phase were the omnidirectionality and the ease of movement, namely having a design that the power of the motor in hand would move easily, which is 7.92W nominal.

The objective that was ruling the flow of design was the omnidirectionality with a realizable mechanical hardware. The number of bandings, orthogonal axes and slot cuts; which are the hardest structures to achieve had to be omitted.

At the first step, the simplest omnidirectional structure was examined. This structure is very similar to differential drive but each wheel in this approach can be steered individually, giving the robot four-degrees-of-freedom. The figure below, Figure 3.1.2 shows the basic schematic representation of such a vehicle, and the one of the simplest wheel steer-drive mechanism that could be achieved using the motor.

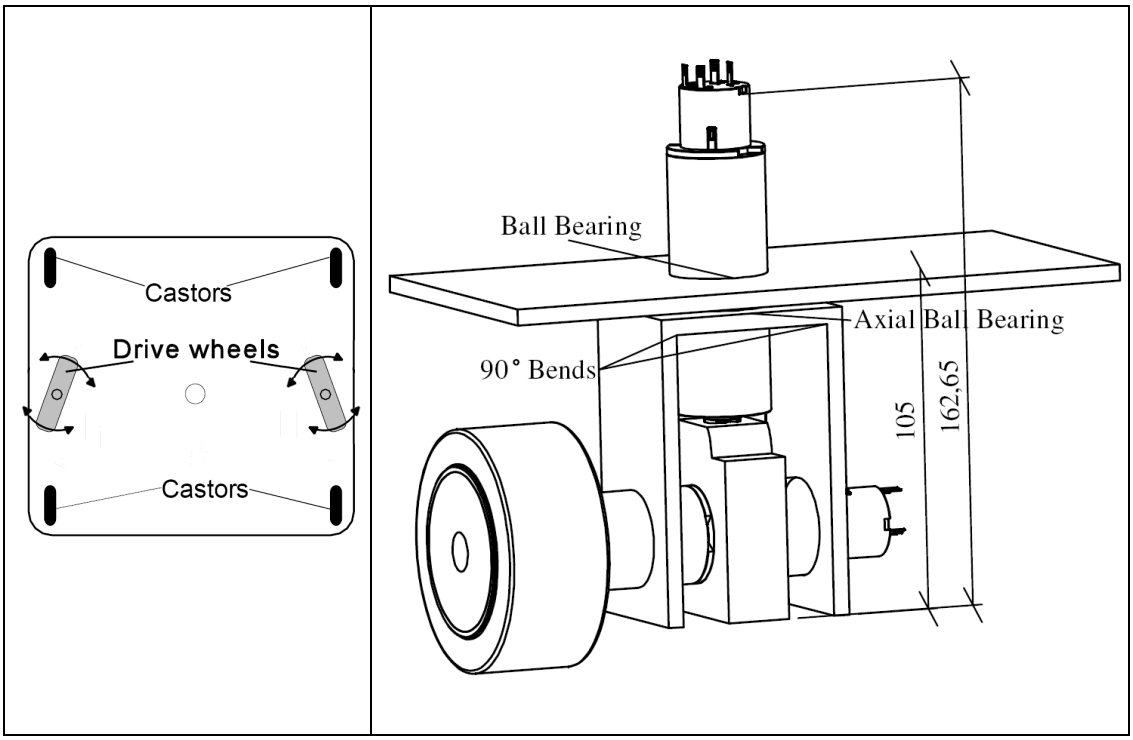

Figure 3.1.2 Classical 4-DOF Device and a Simple Mechanism to Drive Wheels

Even the simplest mechanism designed to drive and steer a wheel at the same time does still include metal bends, axial and rotational bearings, a custom fitting to attach the drive motor to the steer motor and it also requires the six cables of the drive motor to move along with the drive motor as the wheel is steered. Moreover, the closest plane of the platform is so distant to the floor that there cannot be a commercially available castor to support the base on sides; hence the castor would also require custom spacers.

One reasonable solution to the problem is proposed to use two differential drive platforms instead of drive wheels. Differential Drive Platforms (DDPs) are the most common type of robot platforms that uses to wheels or tracks on sides, each one having the ability to be driven forward or backward independently. Hence, to rotate the robot to left, left wheel or truck is operated backwards, while the right moves forward. DDPs are two-degrees-of-freedom devices, since the drive wheels or truck

do not have the ability to be steered. Figure 3.1.3 below shows a simple schematic diagram for a DDP.

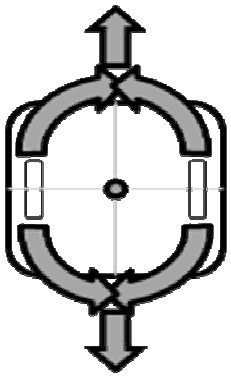

Figure 3.1.3 Schematic Representation of a Differential Drive Platform

The arrows show the possible movement path of the vehicle. The vehicle can move forward-backward and turn around itself, functionally equivalent to the driven and steered wheels required for the MDOF vehicle. As a conclusion the wheels of the MDOF design were replaced by DDPs. Figure 3.1.4 below basically shows the resulting structure.

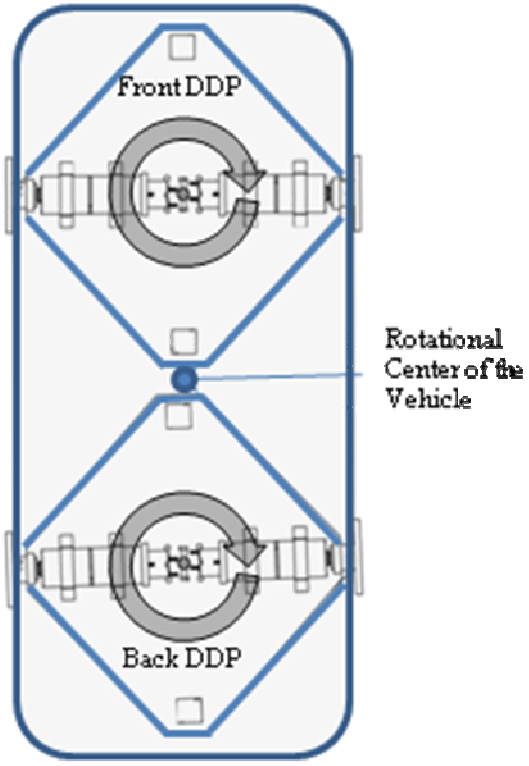

Figure 3.1.4 A MDOF Vehicle Built With Two Differential Drive Platforms

The resulting structure, as it can be understood from the figure above, does not require bending, special fittings, bearings, slot cuts or any kind of hard-to build mechanics. The entire system can be built up with sheet aluminum, simple wheels can be used and motors are simply attached to the surface of the DDP platforms. This solution solved the mechanic complexity of a MDOF design; moreover it provided a large undisturbed sheet metal platform that is capable of moving omnidirectional, which made it easier to fix the batteries, servos and other circuitry on the platform.

After deciding the structure of the platform mechanics, a literature review showed that Borenstein and Evans [10] have proposed a similar mechanics in 1997. They took one step further and made one of the DDPs capable of sliding along the direction of the axis connecting two DDPs. They proved in [6], [21] and [19] that by utilizing the measurement of the compliant link they built, they have reduced the odometry errors successfully. In their work they developed an internal odometry error correction mechanism that uses the sensors from the angles of the DDPs to the robot base and the length of the compliant linkage to compensate the odometry errors. The application of the compliant linkage substitutes an additional correction factor in their odometry correction algorithm. The same algorithm could be used for the design presented in this work, without the improvement made by the compliant linkage. Although it causes additional error in odometry, the compliant linkage mechanism was not added to the design, because it was mechanically hard to implement, a highly accurate linear encoder to measure the link distance could not be found, and among all; the overall tolerance of hand-made mechanics would possibly sum up to make the performance of the platform even worse if a linkage of that kind was implemented.

The final design, based on the principle schematic in Figure 3.1.4 above satisfies omnidirectionality, ease of movement, ease of production and odometrically improvability conditions.

In addition to that, the platform has the capability to be reduced to three-degrees-of-freedom by creating a virtual link between the angles of the DDPs, namely making the second DDP turn to the same heading with the first DDP, as soon as a DDP changes its direction. This makes is possible to reduce the system to a 3-DOF vehicle and overcome if any control problems or inconsistencies are faced, which is always a

possibility [9, 11, 13, 14], during development phase of a tracking algorithm, or an algorithm related to 3-DOF vehicles are being developed. This addition strengthens the design's generality property.

3.2 Details of the Base Mechanical System

In the previous section, it was shown that the proposed mechanics topology satisfies the key objectives of the designs. The further design phase of the mechanics involves the objectives to realize the topology as accurate as possible and make it possible to get needed reading concerning with the mechanics accurately.

Moreover, further in the design, after realization of the first prototype; the need to make the DDPs a little tolerant to movements in the direction of the floor normal, in expense of loosing accuracy has been emerged. The problem shows itself when the robot tries to move parallel to the edge of a two planes with a small difference in altitude, for example when it is moving at the side of a carpet with one side of the platform on the carpet, and the other side is on the floor. When the DDPs are fixed tightly, not allowing bending sideways, the wheels on the floor side do not make enough pressure on the floor and severe slippage of wheels is observed.

Since the structure of the front and the back DDPs are completely the same, only the structure of a DDP is examined in this section. The whole base consists of two symmetrically placed DDPs below a platform of 500mm x 280mm.

The general design considerations require that the DDP must satisfy the conditions below:

a. The drive motors should be fixed in a way such that they should not rotate on their own axis if the wheel is stuck.
b. Since the output shaft of the drive motors are directly mounted to the wheels, the weight of the robot would induce a moment around the shaft's bearing at the output of the motor as shown on Figure 3.2.1 on shaft and possibly damage the motor. The shaft should be supported by a bearing and the vertical lifting forces should be handled by the bearing.

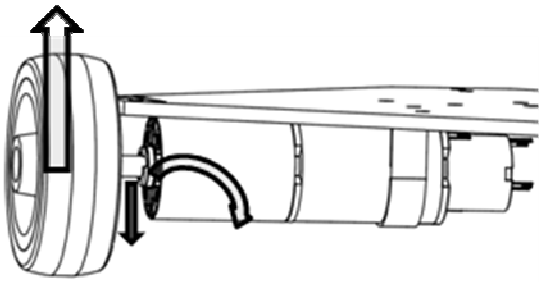

Figure 3.2.1 Induction of Damaging Moments on Motor Shaft

c. The DDP should be rotating freely under the robot's main platform, and carry load at the same time. This could be achieved by using axial ball bearing between two platforms.
d. The DDP should have a tolerance of bending around the central axis. However, this movement should be limited by some sort of castors because of the high moments arising on the central axis of the DDP, where it is mounted to the upper platform.
e. The motor speeds are controlled by "Motor Controller Cards", which need to be close to motors they control, as it will be stated in Section 3.3. The electrical connection of this electronic circuit, which consists of four wires, should be routed to the upper platform. There should be a central shaft, which is a hollow pipe allowing the cables pass through while allowing the DDP to turn around.
f. The angle between the DDP and the base should be measured. A multi-turn potentiometer is used. The potentiometer should be mounted in such a way that the turning of the DDP should turn the potentiometer, preferably with a scaling factor which is greater than one to improve accuracy.

The disassembly schematic of a DDP is given in Figure 3.2.2 which achieves given goals.

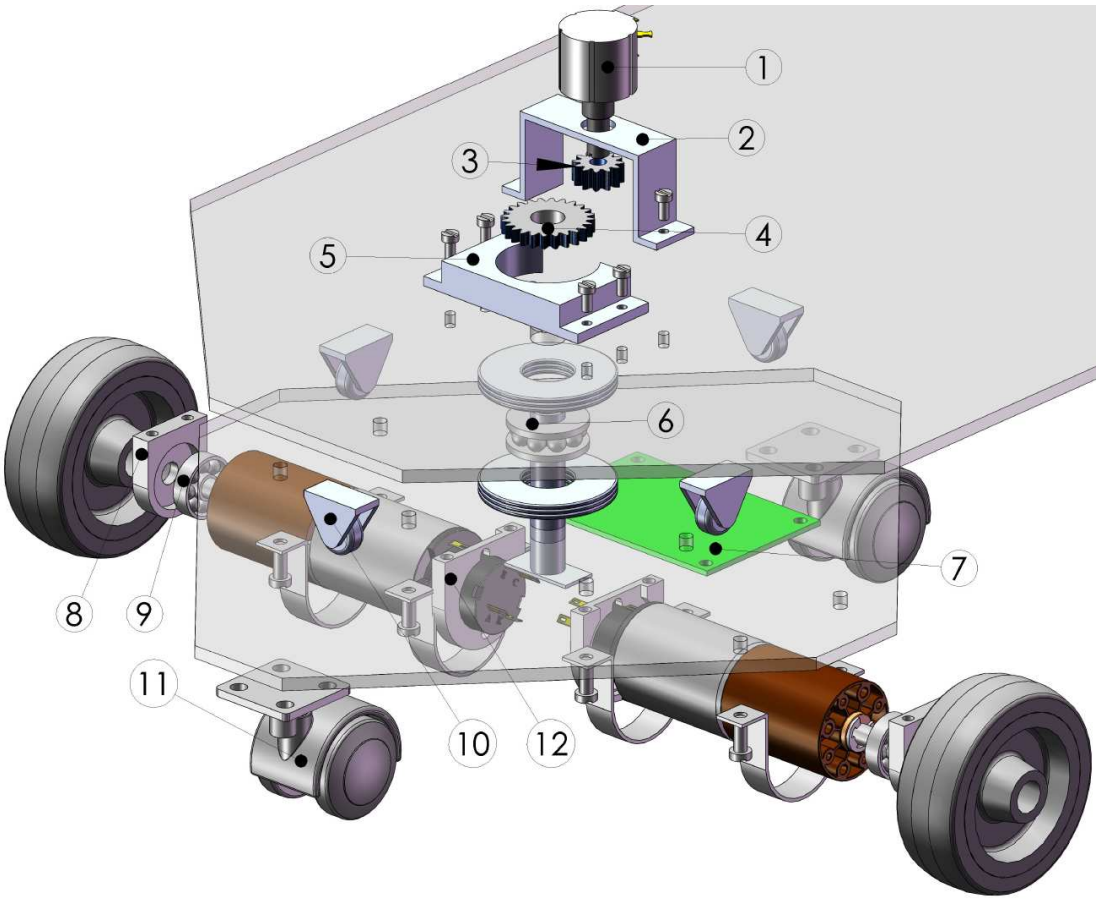

Figure 3.2.2 Disassembly Schematic of the DDP design

The table below, Table 3.2.1 shows the names and functions of the functionally important items that have been numbered on Figure 3.2.2.

Table 3.2.1 Descriptions of Important Components of the DDPs

Item Number	Description
1	A 5 turns wire wound precision potentiometer for measuring the angle of the DDP.
2	A fitting the match potentiometer gear (3) with the axis gear (4).
3	The potentiometer gear. A silicon gear fitting tight on the potentiometer shaft.
4	The axis gear. The axis gear is mechanically connected to the axis pipe of the DDP and reflects the turning of the DDP to the fixed potentiometer over the potentiometer gear.
5	Axle gear fitting that preserves axle gear from disconnecting the potentiometer gear with a bending motion of the DDP.

6	A roller thrust that mechanically isolates the DDP from the base, which allows the DDP to turn freely under the base. This part is chosen to be a roller type; hence the upper and lower surfaces of it can move freely enabling bending movements.
7	The “Motor Controller Card” dedicated to this DDP.
8	A custom fitting that houses the roller bearing (9) and transmits the force on the wheels to the DDP chassis, solving the problem with moments that could harm the motor shaft.
9	A roller bearing that enables the motor shaft to run freely, while the force is transmitted over the bearing and the custom fitting to the chassis.
10	Limiting wheels limit the magnitude of bending of the DDP around the central axis, hence assuring that the system will not be forced to limits.
11	Standard castors prevent the whole robot from falling aside when the four wheels of two DDPs are all in line, which is not unusual in MDOF vehicles.
12	A custom fitting that tightly fits on to the cross-section of the DC motor and the encoder that prevents the motor from turning around itself if the wheel is stuck

Figure 3.2.3 below clarifies how the DDP bends around its central axis and how this motion is limited as required, by giving the overall picture.

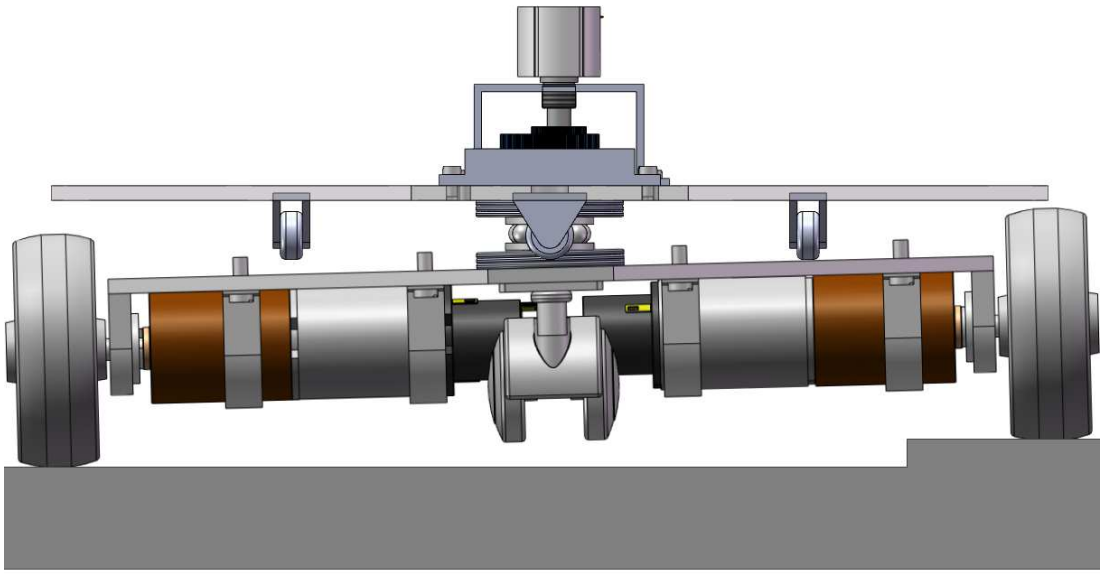

Figure 3.2.3 Bending of the DDP and Limiting of the Action

3.3 Motor Controller Circuits

3.3.1 Motor Controller Hardware

The Each DDP has its own motor driver circuit on its chassis as shown with number 7 on Figure 3.2.2 above. There are three basic reasons for providing separate motor driver cards on each DDP, close to the drive motors:

a. Reducing the number of wires required to be connected to the base controller card on the main platform, through the rotating DDP axis.

   The Motor Controller Cards require four wires, two of which are for the power supply and the remaining two is for serial communication. If the motor control functions were implemented on the main body, at least eight wires would be passing through the axis, which are the two encoder data wires, four wires to power motors as pairs for each motor, a drive supply for the GaAs LED in the incremental encoders and a ground connection. Since the DDP is rotating with respect to the main base, with increasing number of interface wires, and hence the total wire diameter; the possibility for the wires to get stuck inside the rotation axis or to the gears increases. Even with four wires, the total turning dynamic of each DDP is restricted to 360˚, because of the stress accumulating on the twisting interface cables.

b. As it will be presented below, the power of the drive motors are controlled by Pulse Width Modulation (PWM) technique. The motor power is switched on and off at a frequency of 30 kHz and a variable duty cycle that controls averages to the power of the motor. Since the drive motors are power full elements, switching such high power from a distant location and carrying high frequency – high power motor lines through the system would probably induce serious Electromagnetic Interference (EMI) related hardware problems.

c. The speed of a motor is measured with an optical incremental encoder that is fixed behind the motor. The encoder provides the open collector of a photo-transistor that is saturated if the infrared light from the GaAs LED inside the encoder, behind the encoder wheel reaches it. Figure 3.3.1 below shows the structure of a typical optical encoder, similar to ones that are present in the drive motors of the system.

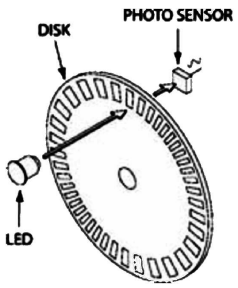

Figure 3.3.1 The Structure of an Optical Incremental Encoder

The data provide by the encoder in logic levels would have to travel on wires parallel with the motor drive wires to a distant location which would clearly influence the weak signal on the encoder sense lines causing inefficiency or even a complete misreading situation.

The motor control tasks of two DDPs are divided and the required circuitry is provided near the motors on the DDP for the reasons stated above.

The electronic structure of Motor Control Cards is quite simple. An eight bit microcontroller, the VRS1000 from Goal Semiconductors, which is a speed, peripheral and memory enhanced 8052 [32], runs the control algorithms. The motors are driven through the PWM channels of the microcontroller with a L298 [33], which is a dual H-Bridge motor driver capable of driving four amperes to each motor at twenty volts. The H-Bridge structure allows the direction control and speed braking of the motors to be done via logic signals. The encoder data is squelched by passing it through Schmitt-trigger gates, providing clean logic level signals to the microcontroller's interrupt pins. The block diagram of the Motor Controller Card is given in Figure 3.3.2.

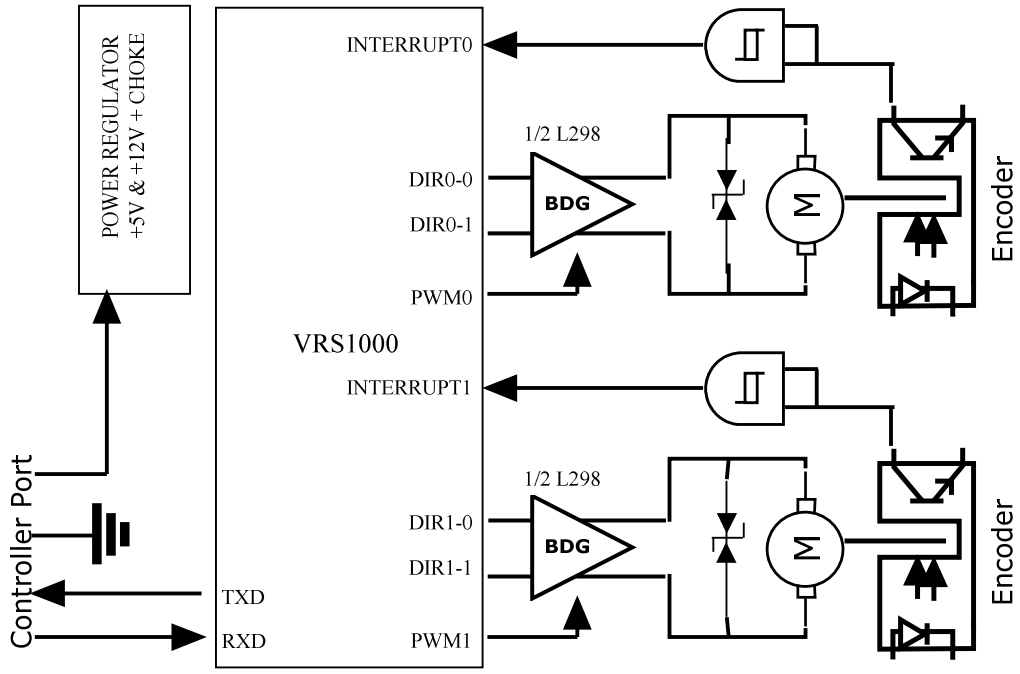

Figure 3.3.2 Block Diagram of the Motor Controller Card

3.3.2 Software Control of Motor Speeds

The software of the Motor Control Card utilizes two discrete Proportion-Integration-Derivative (PID) type control algorithms to drive the motors to given reference velocity value, which is provided by the Base Control Card on the main platform.

The well known PID type control algorithms can be used to control any measurable quantity, that can be changed by modifying other process values [34]. The general transfer function of a PID controller is given as: [35]

$$H(s) = k_P\left(1 + \frac{1}{T_i s} + T_d s\right) \qquad \textbf{(3.3.1)}$$

Where k_p is the gain, T_i and T_d are the integral and derivative time constants respectively. Rewriting this s-domain expression in time domain and multiplying this impulse response with the input signal, namely e(t) which is the error signal obtained by subtracting the actual output value from the reference, yields:

$$u(t) = k_p\left(e(t) + \frac{1}{T_i}\int_0^t e(\alpha)d\alpha + T_d\frac{de(t)}{d(t)}\right) \qquad \textbf{(3.3.2)}$$

This integral and derivative terms in this expression can be approximated to reduce down to the discrete [34] form like in (3.3.3) and (3.3.4) if T denotes the sampling period and n is the discrete step at time t. Which then result in the equation for the controller output given in (3.3.5).

$$\frac{1}{T_i}\int_0^t e(\alpha)d\alpha \approx \frac{T}{T_i}\sum_{k=0}^{n} e[k] \tag{3.3.3}$$

$$T_d \frac{de(t)}{d(t)} \approx \frac{T_d}{T}\left(e[n]-e[n-1]\right) \tag{3.3.4}$$

$$u[n] = k_p e[n] + \frac{k_p T}{T_i}\sum_{k=0}^{n} e[k] + \frac{k_p T_d}{T}\left(e[n]-e[n-1]\right) \tag{3.3.5}$$

The incremental encoders of the drive motors are precision devices that provide 150 pulses per revolution which enables precision odometric information and speed control. The encoders are fed to interrupt pins of the microprocessor as shown on Figure 3.3.2. With each interrupt two accumulators for each motor are incremented, one of which is used for computing the distance travelled by the corresponding wheel, and the other is used for computing the instantaneous speed. With a timer interrupt, that occurs every 20ms, the velocity accumulators are taken as the measurement of velocity at the previous time step and the corresponding PID controller is executed resulting in an update to the corresponding motor power. The velocity accumulators are cleared as soon as they are read and copied to the PID parameters. The Base Controller can read the distance accumulators either with clearing or leaving option, and uses the distance measure for the odometric calculations.

The PID controllers are easy to implement, but the parameters of the controller, namely the gain, integral time constant and the derivative time constant are hard to optimize. The formal method for obtaining the optimal parameters of the PID system is to build a theoretical model of the system and derive the optimal parameters by the analytical procedures available [36]. However, obtaining a complete definitive model of the system is often very hard or even impossible. For this reason, there exists a number of empirical PID tuning mechanisms as well as a great number of stochastic

search based optimization methods. Among the empirical PID tuning methods that were presented in [36, 37], which are the Ziegler-Nichols oscillation method, Ziegler-Nichols tuning by reaction curves and Cohen-Coon tuning; the Ziegler-Nichols oscillation method was chosen because of its simplicity and easy measurement of the parameters needed to obtain tuning input variables.

The Ziegler-Nichols oscillation method is based on a phenomena called quarter-decay criterion [35], and developed mainly from experiment. The method offers setting the proportion and integral terms to zero and increasing the system gain k_{ip} until system runs into a sustained oscillation. Ziegler and Nichols denote critical gain value as the ultimate gain K_U, and the period of the oscillation as the ultimate period T_U and offer a starting point which is believed to be close to optimal values, for an empirical search of optimum system constants [38]. They claim that setting the system gain k_p to $0.6K_U$, integral time constant T_i to $0.5T_U$ and derivative time constant T_d to 0.12Tu would be good configuration to start with. After setting these values, the actual values that provide a plausible performance are found by trimming values.

When the PID controller in (3.3.5) is implemented directly, a problem with integral term arises in certain situations, especially when the motor load is so heavy that even the application of full power is not sufficient to hold the velocity close to the desired value. During these periods large error signals emerge and they are summed up with the integral term until the situation of being stuck is over and the motor is capable again to produce speeds higher than the desired value. However, as the load is reduced back to normal limits and the motor is capable of giving much higher speeds, accumulated large error values by the integral term force the controller to keep applying near full power until the positive errors emerging from the higher motor speed in the case brings the integral term to normal. To overcome this problem the integral term is limited in software. Although this brings up another parameter to define, the effect of this parameter is not practically observable.

3.4 Angle Control of the DDPs

The angles between the DDPs and the robot base provide the most important information in position estimation procedure using odometry, as well as configuring the DDPs according to a given path of movement.

The “Motor Controller Cards” on DDPs are connected to the “Laser Controller and Base Control Card” via their serial ports as it was states previously. The base control card is responsible for turning the DDPs as accurately as possible to the angles given by trajectory controller software running on the “Main Controller Board”. Moreover, the base controller software continuously observes these angles during movements of the robot and corrects the odometric position information by a method that is presented in the next section, Section 3.5.

The base controller uses the same PID approach described in Section 3.3.2 for the motor controller. The only difference in operation is that the main controller commands the required motor control card to drive the required motor to required speed, in contrast to just giving the direction of rotation and the power as the motor controller does. This was achieved by a transformation operation on the control signal output of the PID which includes only an indication of direction and power. The direction information is transformed to two directions of motors of the DDP that will make the DDP turn to the direction required. The power information is scaled and passed as the required velocities of the motors.

Moreover, the Ziegler-Nichols tuning technique [38] that was previously used in tuning of motor controllers’ parameters was not used for tuning base angle controllers. The DDPs are complex systems and turning motion of the DDP is a heavy motion, especially when the weight of robot is considered. Hence, putting DDPs in sustained oscillation by increasing gain was not practical.

As it was stated before, the angles of the DDPs can be observed in real time by a remote host PC. This property was used to tune the parameters of the base angle controllers manually. For each set of parameters tested the angle change of the corresponding DDP was logged and by analyzing the shape of the behavior, the parameters were tuned.

The main parameter that has to be analyzed in this section is the accuracy and resolution of the potentiometer based angle measurement. Figure 3.4.1 shows the mechanical detail and a photograph of the actual construction of the rotational movement transmission between a DDP and the potentiometer responsible of capturing its angle information.

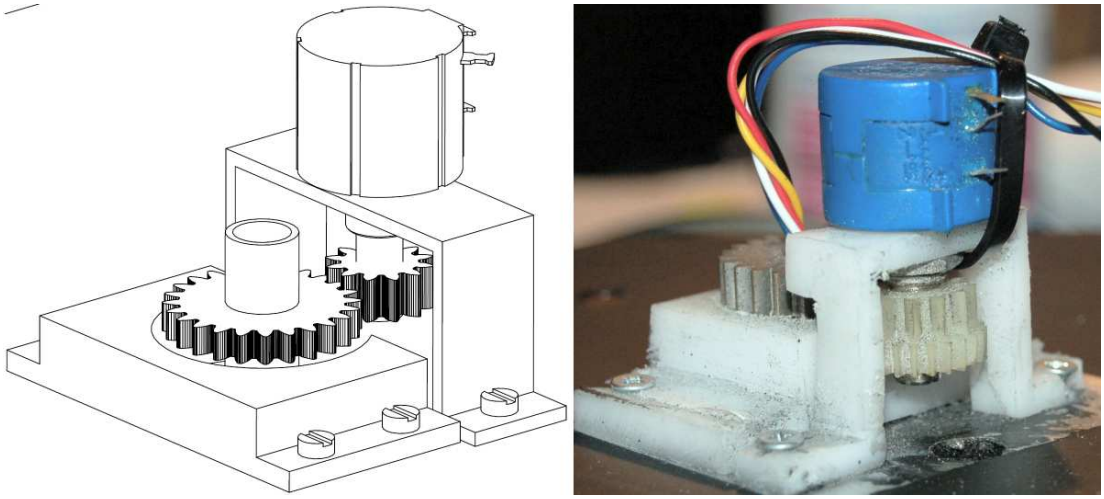

Figure 3.4.1 Mechanical Detail of the Angle Measurement Systems

The ratio of the bigger gear mounted on the rotating axis of the DDP and the smaller gear on the potentiometer is 2/1 giving a two-fold enhancement on the measurement accuracy. However as the total amount of rotation of a DDP is limited to 360˚, the maximum magnitude of the movement of the potentiometer shaft is only 720˚ by the help of gear ratios, where the available dynamic of the potentiometer is actually 1800˚. A solution to overcome this inefficiency and to be able to use the whole dynamic range of the analog to digital converters on the Base Controller Card's microcontroller, the LPC2138 was designed.

The total voltage dynamic range of the Analog to Digital Converters (ADC) of LPC2138 is 3V [39]. To use the whole dynamic range, a 720˚ turn of the potentiometer should vary 3V in magnitude. By adjusting the voltage applied between the static pins of potentiometers, the output dynamic can be adjusted by using (3.4.1) below:

$$\frac{R_U}{R_T} = \frac{\theta_U}{\theta_T} = \frac{\Delta V_U}{\Delta V_T} \qquad \textbf{(3.4.1)}$$

Where R_U denotes the usable range of resistance, R_T is the total resistance between static ports, θ_U is the utilized angle of rotation, θ_T is the total range of rotation, ΔV_U is the resulting dynamic range of the variable port's voltage and ΔV_T is the voltage applied between static terminals. Hence ΔV_T is found as:

$$\Delta V_T = \frac{\Delta V_U \theta_T}{\theta_U} = \frac{(3V).(1800°)}{(720°)} = 7.5V \qquad \textbf{(3.4.2)}$$

As the result in (3.4.2) showed that by using additional voltage regulators which provide 7.5V to these potentiometers the whole dynamic range of the ADC could be utilized, each potentiometer in the design has a linear voltage regulator made with LM1117ADJ [40] close to its static terminals to reduce noise on the measurement line.

The solution obtained above assures the output of the potentiometer will change by 3V with a full 360˚ turn of the DDP which fully fills the dynamic range of the ADC, however the ADC requirement that no offset is present on the measurement line, namely, the input voltage of the ADC should vary between zero and three volts makes it a must to build an interface circuit to cancel the offset present in the potentiometer output. The reason that forces the potentiometer to have an offset is purely mechanical. It was possible to use the first 720˚ of the potentiometer, those are closest to ground and make the output of the potentiometer vary between zero and three volts, however this would be risky; because the potentiometer has a mechanical stopper at the zero point, and the DDP would move down to zero. Any loss of control on DDP angle, an overshoot of angle controller towards negative region or even a bump or an obstacle mechanically forcing the DDP to turn towards negative region would induce excessive forces on the gears since the potentiometer gear is mechanically limited and would be unable to turn, which could result in damaging the gears.

To assure safe operation and compensate for mechanical overshoots of the DDPs' turning action on software limits, the potentiometer turn in their second and third turns, namely their the 360˚-1080˚ region. The output voltage of the potentiometer is calculated as in (3.4.3) in ideal.

$$V_o = \Delta V_T \cdot \frac{\theta}{\theta_T} \qquad \textbf{(3.4.3)}$$

Where θ is the actual position of the shaft. Using (3.4.3) it is known that the output voltage of the potentiometers varies between 1.5V and 4.5V. As it is seen, the dynamic range of 3V is present, but with an offset of 1.5V. To cancel this offset a difference amplifier with MCP6004 opamp from Microchip [41] was built that subtracts 1.5V reference voltage generated by a precision voltage reference from National, LM4041 [42]. The circuit structure is shown is Figure 3.4.2 below.

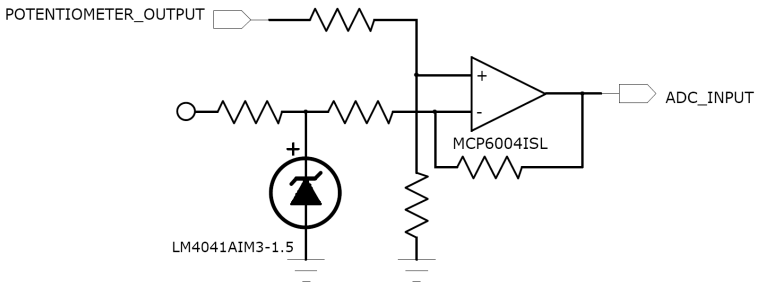

Figure 3.4.2 Structure of the Offset Cancelling Circuits

Although the proposed measurement structure with gears, voltage regulators and offset cancelling circuits solves the transducing of rotary movement of DDPs into analog voltage of required range, they have negative effects on the system accuracy.

The gears mechanically amplify the rotary movement of the DDP by two, but they have inaccuracies due to inevitable gaps between gear threads. These gaps divert the actual gear ratio by a small amount and induce discontinuities of motion which reduce the accuracy and linearity of the measurement. The effect of gear thread gaps is not constant and it's changing with the angle of the DDP both in vertical and in horizontal axis, vibration and even temperature. Building gear systems without thread gaps is not possible in practice, but in order to reduce the amount of divergence the gears used on the potentiometers are made from silicon and the gear couples are placed tighter than it is possible to achieve with metal gears. Hence the silicon gears change their form slightly on the contact area to fill gaps with the larger gear.

The tolerances of passive components in offset cancelling circuits and the tolerance of the measurement potentiometer itself induce measurement shifts and linearity changes that can be assumed to be constant over time, and hence can be eliminated

by calibration process. The total effect of passive component tolerances is not constant in practice; they change with the change of conductivity due to temperature and even to humidity.

Another biggest source of inaccuracy is the operational amplifier used in offset cancelling. MCP6004 has an input offset voltage ranging from -4.5mV to 4.5mV, which changes typically ±2μV for each degree Celsius change in temperature.

Finally, the resolution of the ADC is another important parameter. The resolution of LPC2137 ADC channels is 10bits, which means that the dynamic range of 3V is quantized into 1024, which brings the reading resolution of about 2.93mV. discarding the inefficiencies of system components, the reading achieved by software for DDP angle θ is computed as in (3.4.4) where C is the digital conversion result, θ_{OFF} is the offset of 360˚ in the potentiometer, N_P is the number of threads on the potentiometer gear which is 12, N_D is the number of threads on the DDP gear which is 24, V_{REF} is the generated reference voltage of 1.5V at the offset canceller and Q is the number of quantization levels which is 1024.

$$C = \left[\left[\left(\theta \frac{N_D}{N_P \theta_T} + \theta_{OFF} \right) \frac{\Delta V_T}{\theta_T} \right] - V_{REF} \right] \frac{Q}{\Delta V_U} \qquad \textbf{(3.4.4)}$$

Clearly, the ideal resolution of the angle measurement system is then 0.352˚. By considering the inefficiencies, (3.4.4) may be rewritten as in (3.4.5).

$$C = \left[\left(\left(\theta \left[K_G \frac{N_D}{N_P} \right] + \theta_{OFF} \right) \left[\frac{1}{\theta_T + C_P} \right] \right) \left(\Delta V_T . K_T + C_T \right) - \left(V_{REF} . K_{OCC} + C_{OCC} \right) \right] \frac{Q}{\Delta V_U} + C_{AD} \qquad \textbf{(3.4.5)}$$

K_G represent the effect of gear thread gaps on gear ration, C_P is the tolerance of actual range of the potentiometer, K_T and C_T are parameters induced from the tolerance of LM1117ADJ voltage regulator, K_{OCC} and C_{OCC} are induced from the tolerances of the passive and active components of the offset canceller and C_{AD} is the conversion tolerance of the ADC. The tolerance influencing ΔV_U was not included, because the potentiometer reference voltage regulator is fed from the same source as the ADC, hence changes on this voltage is not effective on the conversion result.

To obtain an approximation for the worst case accuracy, values of tolerance components are estimated. The effect of gear thread gaps is neglected. The tolerance of the potentiometer is known to be 1%, giving a C_P of ±9˚. The maximum load regulation error of LM1117ADJ is given as 0.4% and the offset error is %2 giving K_T equal to 1 ± 0.02 and C_T equal to ±7.5mV. The accuracy of the reference voltage is given to be %0.3 over industrial temperature range and the offset drift of the MCP6004 is ±2μV, concluding an approximation for K_{OCC} as 1 ± 0.003. C_{OCC} is mainly resulting from the input offset voltage of MCP6004 which is between -4.5mV and 4.5mV, giving a C_{OCC} of ±4.5mV. Finally, C_{AD} is mainly from the absolute error of the ADC which is given to be ±4.5 least significant bits. From (3.4.5) it is easy to see that the maximum error will occur when θ is at maximum, namely at 360˚, and polarities of tolerances are in arrangement those induce errors at the same direction, not cancelling each other. Evaluating (3.4.5) for this value with tolerance gives an error of 33.82 which makes 3.3% of error at maximum. However the effect of initial input offset voltage of MCP6004 C_{OCC} and the offset error of LM1117ADJ C_T could be reduced by properly trimmed component values. Moreover the effect of C_P could also be reduced by adjusting ΔV_T accordingly. Hence the worst case error rate converges approximately to 0.9%.

3.5 Positioning System Details

The positioning system of the base relies mostly on dead-reckoning, namely odometry. The properties and advantages of this approach were discussed in Section 1.2.1 “Movement Flexibility, Speed and Accuracy”.

This section introduces the main sources of errors in the odometric positioning system and proposes to approaches to reduce errors.

In Section 3.5.1, type types of odometry errors and sources of the odometry errors are discussed. The next section, Section 3.5.2 presents a solution to reduce odometry errors by utilizing the advantages of the mechanical design which were previously discussed.

3.5.1 Odometry Errors

Odometry errors can be grouped into two, the systematic error and non-systematic errors.

Systematic odometry errors are the ones that are specific to the robot and do not change significantly throughout the movement. These errors are mainly results of difference of the actual wheel diameters and the tolerance of the effective wheel base [43].

The wheelbase is defined as the distance between the contact points of the wheels to the floor surface in a differential drive robot [21]. The wheelbase value is used to determine the difference of distances taken by the differential motors to give the robot a certain amount of rotation. The uncertainty arises from the fact that, the contact between the wheels and the floor surface is actually an area rather than a point. The effect of uncertainty of the wheelbase shows itself only when the robot is turning.

The wheel diameters are not exactly equal in practice. Many robot platforms, including the one presented in this work, use rubber tires to prevent slippage and excessive deformation of wheels. Rubber is an elastic material, which makes it hard to produce equal diameter wheels; moreover this elasticity allows rubber to deform and change the effective diameter of the wheel according to the weight loaded onto the wheel. Using metal wheels significantly increase the probability of slippage on smooth surfaces. Using plastic wheels also increase the slippage rate in comparison to rubber tires, but the main problem in plastic wheels is the deformation of tire due to friction. These results make it advantageous to use rubber wheels appropriating their disadvantages. Nevertheless, reducing the effects stated about the rubber wheel is possible. In this work, the wheels of robot were constructed combining two parts. Precision internal wheels were produced using a computer controlled modeling machine, which were made of Delrin, which is one of the most durable plastic substrates available. The diameters of the internal wheels are 60mm. After producing the internal wheels, covers made of hard black rubber were attached increasing the wheel diameter to 65cm. Thin ribbons of hard rubber on hard plastic internal wheels prevent the wheel from slipping and deformation while being immune to the weight of robot and not contradicting to change wheel diameters. This reduces the wheel

diameter problem to a completely static problem, which can be compensated in software by calibration. Without calibration, the unequal wheel diameters result in a curved path when the robot is ordered to move in a straight line.

An experimental way exists to calibrate the wheel diameters of a differential drive robot [18]. The vehicle is programmed to go on a straight line for a number of runs. In each run, the robot would complete a curved path instead of a straight line as needed due to wheel diameter differences. At the end the average radius of the curved path is measured.

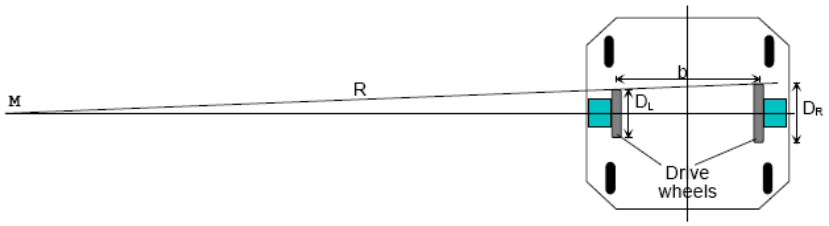

Figure 3.5.1.1 Parameters Needed to Calibrate Wheel Diameters

With the wheelbase b, designed wheel diameter D_L and D_R for left and right wheels, and the measured radius of curvature R is known, the actual ratio of wheel diameters E_d can be computed by the equation (3.5.1.1) below [18].

$$E_d = \frac{D_R}{D_L}\frac{R+\left(\frac{b}{2}\right)}{R-\left(\frac{b}{2}\right)} \qquad \textbf{(3.5.1.1)}$$

While deriving an expression for the correction factor, the average wheel diameter is kept constant in order to make sure that recalibration of it will not be required. Once E_d is computed, the corrected values of actual wheel diameters, namely the corrected value of left wheel diameter D_{LC} (3.5.1.2) and the corrected value of the right wheel diameter D_{RC} (3.5.1.3) can be found [43].

$$D_{LC} = \frac{2}{E_d+1}\frac{\left(D_R+D_L\right)}{2} = \frac{D_R+D_L}{E_d+1} \qquad \textbf{(3.5.1.2)}$$

$$D_{RC} = \frac{2}{\left(\frac{1}{E_d}\right)+1}\frac{\left(D_R + D_L\right)}{2} = \frac{E_d\left(D_R + D_L\right)}{E_d + 1} \qquad \textbf{(3.5.1.2)}$$

Although this calibration method is defined for differential drive robots, the initial calibration of the MDOF design presented here can be done using the same technique for each DDP at a step. Since a single DDP is strong enough to move the robot by pulling the other DDP and a slip-free run was assured by realizing the test on a silicon coated floor, that was originally back side of an a antistatic coating used in electronics. Each DDP was calibrated by turning off the other DDP and applying the calibration method above as if the robot is a single differential drive robot.

Compensation of systematic odometry errors is generally possible. The main problem sources dealing with the efficiency of odometry are the non-systematic errors.

Non-systematic errors are unpredictable momentary errors caused by mostly the irregularities of the floor or the slippage of wheels. Irregularities like cracks, bumps and holes on the floor cause one or more of the wheels make a displacement that is not induced by motor, and wheel slips cause the true position of the wheel diverge from the mathematically predicted position constructed on erroneous encoder data. Non-systematic errors are not correctable in most of the conventional approaches. [21].

3.5.2 Correction of Odometry Errors

The mathematical approach proposed for odometry error reduction in the design is an application of the Internal Position Error Correction (IPEC) algorithm which was developed by Borenstein [10, 21]. Although the IPEC algorithm was developed for using with a compliant linkage robot, the procedure is still capable of reducing odometry errors.

The main inspiration of the algorithm is a similar to the multi-robot approach presented in the works of Karazume and Nagata [44]. This approach used two groups of robots. Robots in one of the groups stay constant whenever the other group is in motion, hence enabling the moving robots to see and correct their odometry errors with taking the second group members as reference points. Similar to this, if DDPs

of the robot are considered to be distinct robots, detection and correction of odometry errors of robots by referencing each other is the main idea.

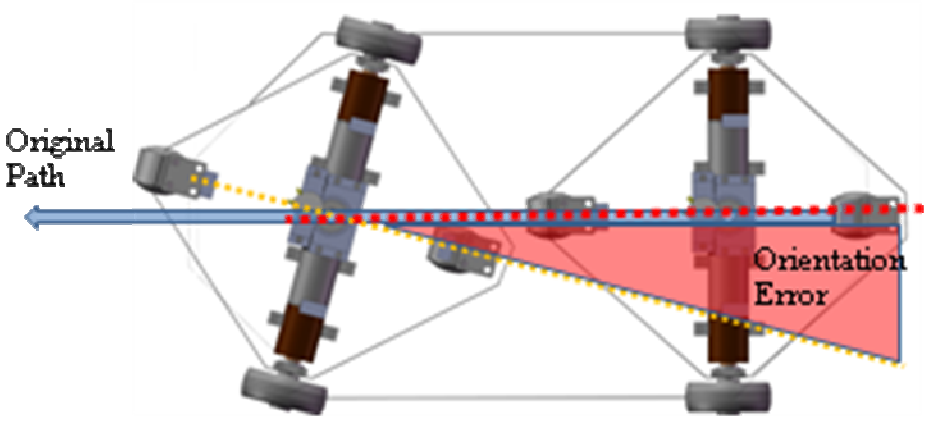

Figure 3.5.2.1 Occurrence of an Orientation Error

Figure 3.5.2.1 shows a typical orientation error situation where the front DDP is disoriented by some external reason. These reasons might be anything like a bump that the right wheel had to travel over, a slip of the right wheel or a collision of the right side of the front DDP with a small obstacle that could not be caught by the avoidance system. In this situation, the information retrieved in means of odometry, namely from the number of counts that encoders still state that the front DDP is in line with the back DDP. The base controller calculates the relative angle between the front truck and the back truck compares it to what it has foreseen by the odometrically calculations and finds the orientation error angle. Basically, the front DDP looks for the back DDP which he expects to see right behind, and finds it misaligned. At this step it is not known that if the front DDP has faced and error and turned right, or the back DDP has faced an error and turned left; but while repeating this expectation-comparison process for the back DDP, the base controller finds out that the back DDP estimates the relative bearing of the front DDP correctly, concluding that the front side is erroneous. Moreover the direction and the magnitude of error are also known. Hence the error in odometric calculations can be corrected.

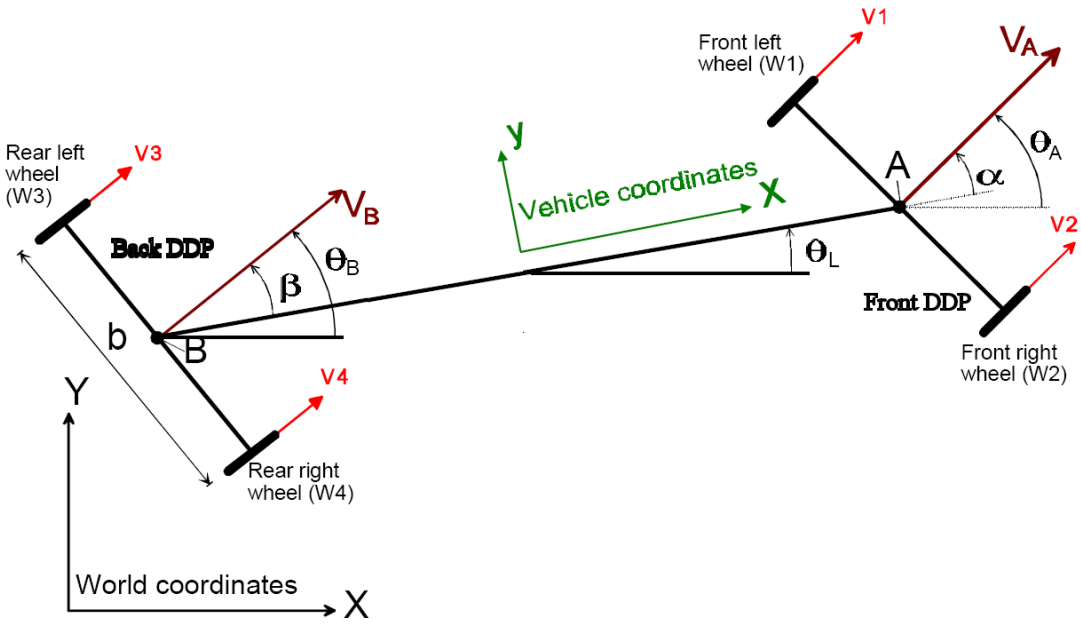

Figure 3.5.2.2 Kinematic Definitions

To explain the working mechanism of the correction mechanism, definitions shown in Figure 3.5.2.2 has to be made. The real world coordinates of the front and back DDP centers, which are shown s points A and B in the figure above, at X and Y directions respectively, and the orientations of DDPs are calculated from basic dead-reckoning equations below, where T is the sampling period.

$$X_A[n] = X_A[n-1] + b\frac{V_1+V_2}{2(V_2-V_1)}\left[\sin\left(\frac{T(V_2-V_1)}{b}+\theta_A[n-1]\right)-\sin\theta_A[n-1]\right] \quad \textbf{(3.5.2.1)}$$

$$Y_A[n] = Y_A[n-1] - b\frac{V_1+V_2}{2(V_2-V_1)}\left[\cos\left(\frac{T(V_2-V_1)}{b}+\theta_A[n-1]\right)-\cos\theta_A[n-1]\right] \quad \textbf{(3.5.2.2)}$$

$$\theta_A[n] = \theta_A[n-1] + \frac{T(V_2-V_1)}{b} \quad \textbf{(3.5.2.3)}$$

$$X_B[n] = X_B[n-1] + b\frac{V_3+V_4}{2(V_4-V_3)}\left[\sin\left(\frac{T(V_4-V_3)}{b}+\theta_B[n-1]\right)-\sin\theta_B[n-1]\right] \quad \textbf{(3.5.2.4)}$$

$$Y_B[n] = Y_B[n-1] - b\frac{V_3+V_4}{2(V_4-V_3)}\left[\cos\left(\frac{T(V_4-V_3)}{b}+\theta_B[n-1]\right)-\cos\theta_B[n-1]\right] \quad \textbf{(3.5.2.5)}$$

$$\theta_B[n] = \theta_B[n-1] + \frac{T(V_4 - V_3)}{b} \quad \textbf{(3.5.2.6)}$$

After computing the expected positions and headings of the DDPs from the equations above, the angle of the robot body, shown as θ_L in Figure 3.5.2.2 is computed as in equation (3.5.2.7).

$$\theta_L[n] = \arctan\left(\frac{Y_B[n] - Y_A[n]}{X_B[n] - X_A[n]}\right) \quad \textbf{(3.5.2.7)}$$

By using θ_L and the DDP orientations θ_A, θ_B the expected values of the angles between DDPs and the robot body α_{exp} for front DDP and β_{exp} for back DDP can be computed.

$$\alpha_{\exp}[n] = \theta_A[n] - \theta_L[n] \quad \textbf{(3.5.2.8)}$$

$$\beta_{\exp}[n] = \theta_B[n] - \theta_L[n] \quad \textbf{(3.5.2.9)}$$

α_{exp} and β_{exp} values are the angles that are expected to be measured by the potentiometers on points A and B, measuring the DDP angles with respect to the robot body. Any odometry error in the previous time interval makes the expected values differ from the actual readings α and β acquired by angle measurement hardware. Thus, the amount of rotational odometry errors made for orientations of DDPs θ_A, θ_B can be computed as in equations (3.5.2.10) and (3.5.2.11) and the corrected orientations θ_{ACR}, θ_{BCR} to be used for the next time interval's calculations can be corrected by adding differences.

$$\Delta\theta_A = \alpha - \alpha_{\exp} \quad \textbf{(3.5.2.10)}$$

$$\Delta\theta_B = \beta - \beta_{\exp} \quad \textbf{(3.5.2.11)}$$

$$\theta_{ACR} = \theta_A[n] + \Delta\theta_A \quad \textbf{(3.5.2.12)}$$

$$\theta_{BCR} = \theta_B[n] + \Delta\theta_B \quad \textbf{(3.5.2.13)}$$

By correcting $\theta_A[n]$ to θ_{ACR} and $\theta_B[n]$ to θ_{BCR} for the calculations of $\theta_A[n+1]$ and $\theta_B[n+1]$ the rotational odometry errors made in the sampling interval are compensated.

The method explained above detects and corrects rotational errors. However, in general case errors caused by floor irregularities and wheel slippage are composite errors, meaning that they cause translational errors as well as rotational errors. As a method for correcting rotational error is ready, to correct the translational error, the only information needed is the point around which the rotation occurred. A result stated in [45], claims that an orientation error is always caused by an encoder reporting a distance longer than the actual distance travelled by the wheel, in practice. This result makes it possible to determine the point of rotation. As an example, the point of rotation in Figure 3.5.2.2 is the front left wheel because the front left wheel has lagged behind the right wheel, hence the encoder of front left wheel reported a longer distance than the wheel has taken in horizontal direction and calculations erroneously yielded that the two wheels should be in line with the direction of the movement.

Once the DDP causing the error, the rotational error and the point of rotation are successfully determined the translational error of the DDP can be corrected using equations (3.5.2.14) and (3.5.2.15).

$$X_{ACR} = X_A[n] - b\sin\left(\frac{\Delta\theta_A}{2}\right)\cos\theta_{ACR} \qquad \textbf{(3.5.2.14)}$$

$$Y_{ACR} = Y_A[n] - b\sin\left(\frac{\Delta\theta_A}{2}\right)\sin\theta_{ACR} \qquad \textbf{(3.5.2.15)}$$

Similar to the rotational error correction, by correcting $X_A[n]$ and $Y_A[n]$ to X_{ACR} and Y_{ACR}, the translational error induced on front DDP in the sampling period is compensated. The Figure 3.5.2.3 below shows the occurrence of the translational error due to lag of left wheel of the front DDP in the example case.

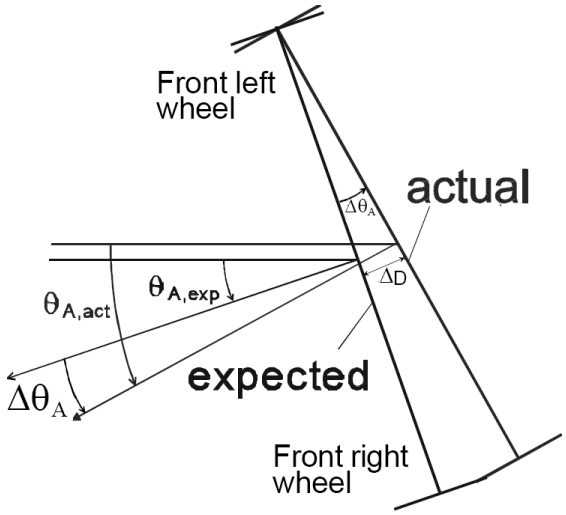

Figure 3.5.2.3 Translational Error due to Lag of a Wheel

Before performing computations for the next period, another critical correction is made on the values. Since the distance L between points A and B, namely the rotation axis centers of the DDPs, on the robot chassis is known; the coordinates of – DDP whose coordinates were not modified are changed to match to the known distance L. In the example in consideration, the coordinates of back DDP are corrected as given in equations (3.5.2.16) and (3.5.2.17).

$$X_{BCR} = X_{ACR} - L\cos\theta_L \qquad \textbf{(3.5.2.16)}$$

$$Y_{BCR} = Y_{ACR} - L\sin\theta_L \qquad \textbf{(3.5.2.17)}$$

The values of $X_B[n]$ and $Y_B[n]$ are modified to X_{BCR} and Y_{BCR} before the next cycle. This last correction prevents changing of θ_L due to accumulation of errors, which is inevitable.

4. ENVIRONMENTAL SENSE

Every mobile robot platform must have sensors the gather information about the environment it is moving within, to avoid obstacles and successfully complete its tasks concerning with environment.

There are a numerous types of sensors for environmental sensing. Ultrasonic and infrared sensors, tactile sensors and laser rangers are the most common types. Various advantages and disadvantages of these kinds of sensors were stated in Section 1.2.2.

An ideal sensor for environment definition should be capable of measuring distance to a point of interest with reasonable accuracy and directivity. Moreover, the operational range of the sensor should be sufficient to map a relatively large area. Among available proximity or range sensors, the laser range meters are the most suitable according to these definitions. However, other aspects of design, such as cost and simplicity limit usage of these kinds of sensors. There are a number of types of laser distance measurement systems available.

Time-of-flight type of laser range meters measure the time between the release of the laser beam and the detection of the reflection from the target's surface. Since the light takes approximately 30cm per second, the timing electronics needed should be capable of measuring a 50ps delay to achieve accuracy about a centimeter [46], which makes this kind of systems very expensive.

An alternative to time-of-flight type is the pulse-shift laser range meter which measures the phase shift between the outgoing amplitude modulated laser beam and its reflection. This approach allows dealing with lower frequencies, since only the modulation signal is concerned which is typically around 15MHz. However, to measure the phase shift complicated electronics including modulators, oscillators and mixers as well as custom optics like semi-reflective mirrors are needed [23].

Every type of laser range meter includes custom lenses, ultra fast response avalanche photodiodes and high voltage generators to drive avalanche diodes; which make them unsuitable to adapt to custom designs.

In this work, a range meter that utilizes a simple laser pointer, which is simply turned on or off without modulation; and a video camera, which is already a part of the design was implemented. The main idea giving the principle idea was found on a web page of "UB Robotics" [47], and another page [48] referencing the previous. The system is capable of approximating the distance of any point on three dimensions which is the field of view of the system camera.

Main idea behind the concept is that the image of a laser point gets closer to the horizon, which is the mid-line of the camera as it gets far. Inversely, the image of the laser point gets closer to the image border as the actual distance of the laser point gets closer to the camera. The Figure 4.1 below shows an example photograph which shows this phenomenon over a laser line.

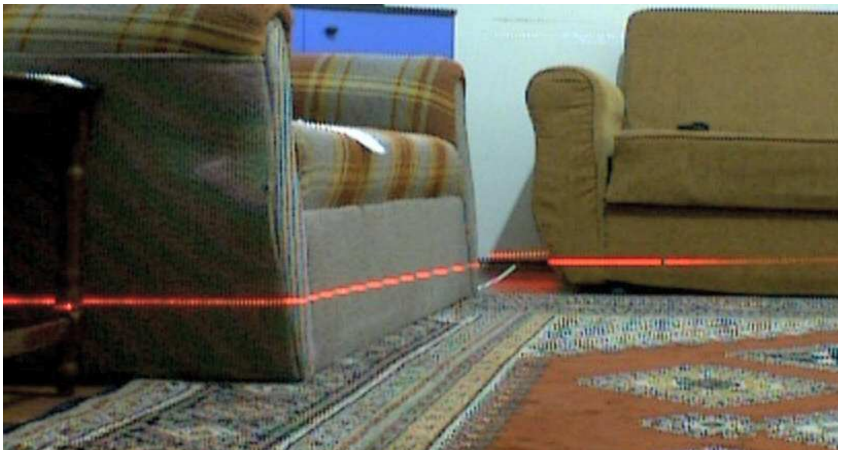

Figure 4.1 An Example of Changing of Laser's Location on Image with Distance

Section 4.1 provides the theoretical information about the implementation the laser range measurement system. The basic idea behind the theory is presented and the mathematical relations between the real coordinates and the image coordinates of a laser point are shown

The next section, Section 4.2 defines the image processing procedures needed to determine the point on the image accurately is presented.

4.1 Theory of Operation

Both the camera and the laser source are capable of moving in pitch and yaw axes, which makes a wide range of operation possible. However for simplicity, it is convenient to start with a situation where the laser and the camera are parallel to determine system behavior and then generalize the results to any orientation. Figure 4.1.1 below shows a diagram of such a system.

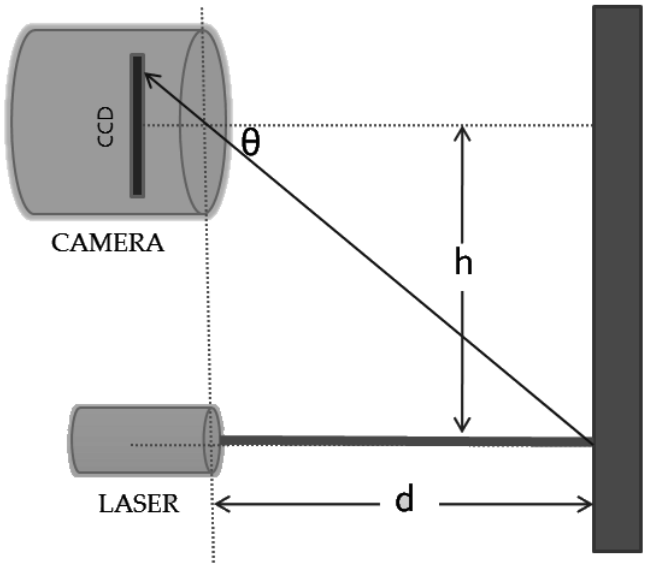

Figure 4.1.1 Basic Diagram of Operation

Let C_Y be a constant denoting the change in angle θ for each pixel pitch on the CCD, which will be derived from the CCD image sensor size, resolution and the focal length of the camera later in this chapter. If y is vertical the coordinate of the image of the laser point on the CCD and R_V is the vertical resolution of the CCD, then the value of θ can be derived from

$$\theta = \left| y - \frac{R_Y}{2} \right| C_Y \qquad \textbf{(5.1.1)}$$

After obtaining θ, since the distance h between the camera and the laser is known, it is easy to obtain the distance d using the trigonometric relation between θ, h and d.

$$d = \frac{h}{\tan\theta} = \frac{h}{\tan\left(\left| y - \frac{R_Y}{2} \right| C_Y \right)} \qquad \textbf{(5.1.2)}$$

A similar expression to obtain the unknown distance d can be obtained easily even if the camera and the laser axes are not parallel.

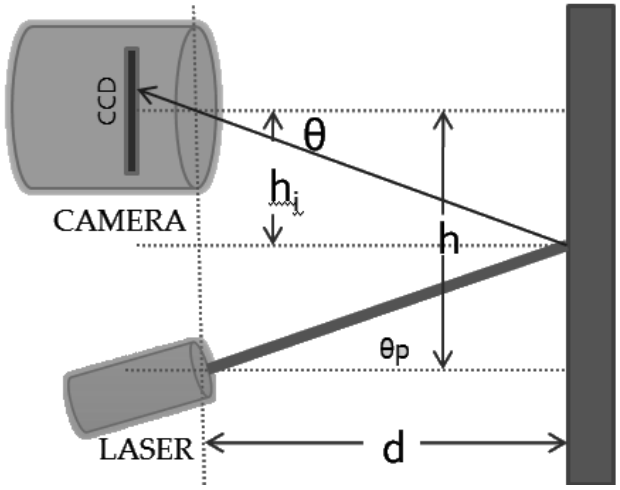

Figure 4.1.2 Operation with Non-parallel Axes

Equation 4.1.1 given for θ is not dependent on axial orientation; hence θ and h are known variables. The angle θ_P is the angle of the laser beam from the direction of the camera axis. Since the camera and the laser are both controlled by precision servo systems, the θ_P is a known angle that is equal to the angle of the pitch servo motor. With the known variables, one can write

$$d = \frac{h_i}{\tan\theta} = \frac{h - h_i}{\tan\theta_p} = \frac{h}{\tan\theta\left(1 + \frac{\tan\theta_p}{\tan\theta}\right)} \qquad \textbf{(4.1.3)}$$

$$d = \frac{h}{\tan\left(\left|y - \frac{R_Y}{2}\right| C_Y\right) + \tan\theta_P} \qquad \textbf{(4.1.4)}$$

If the camera also has an inclination from the parallel direction, which would be the angle of the camera pitch motor similarly, clearly this angle should be subtracted from θ, assuming that the pitch angle is zero at the parallel position with the floor surface, positive if pitched up and negative if pitched down. Namely, if the pitch angle of the camera is θ_T, the most general form of the distance formula for vertical axis can be obtained as

$$d = \frac{h}{\tan\left(\left|y - \frac{R_Y}{2}\right| C_Y - \theta_T\right) + \tan\theta_P} \tag{4.1.5}$$

The same approach can be used for the horizontal axis. Symmetrically, if x is the horizontal coordinate of the laser spot image on CCD, R_x is the resolution of the CCD in horizontal axis and C_x is the coefficient showing radians per pixel pitch in horizontal direction, θ_Y is the angle of the yaw servo of the laser and θ_L is the angle of the camera servo, then the distance formula for the horizontal axis in general form would be as in Equation 5.1.6. The parameter w is the distance between the laser and the camera in horizontal axis.

$$d = \frac{w}{\tan\left(\left|x - \frac{R_X}{2}\right| C_X - \theta_L\right) + \tan\theta_Y} \tag{4.1.6}$$

The calculations in horizontal and vertical axis should give the same distance value for the same image in ideal, but in practice these values differ due to tolerances of the servo angles. Hence, when estimating the distance of a point two calculations are averaged to give an approximation.

The equations above provide the distance of the laser point's projection to the plane defined by the fundamental axis of the camera and the ground normal. Figure 4.1.3 below gives a look to measurement of a distance from the top of the robot.

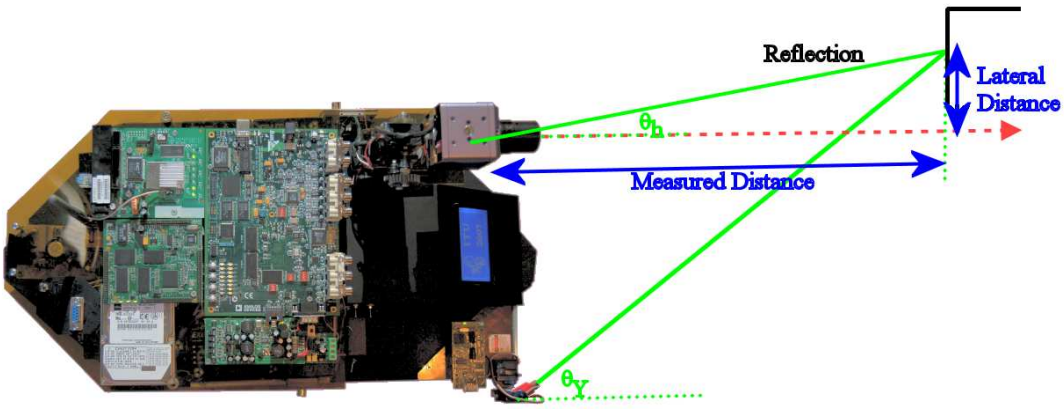

Figure 4.1.3 Definition of the Measured Distance

Another important distance that has to be calculated in order to provide a map of the environment is the lateral distance d_L defined on Figure 4.1.3. By having a method for calculating lateral distance, it is possible to build a map of the environment without needing to move the camera, by just scanning with the laser. The lateral distance can be calculated in two ways. The first way is to calculate θ_h value by

$$\theta_h = \left| x - \frac{R_X}{2} \right| C_X \qquad \textbf{(4.1.7)}$$

And then with the distance value d calculated, the lateral distance can be found by

$$d_L = d \tan\theta_h = d \tan\left(\left| x - \frac{R_X}{2} \right| C_X \right) \qquad \textbf{(4.1.8)}$$

The second way to calculate the lateral distance is to use the trigonometric relation between d_L the lateral distance, w the horizontal distance between laser and camera, θ_Y the angle of the laser yaw servo and the measured distance as

$$d_L = d \tan\theta_Y - w \qquad \textbf{(4.1.9)}$$

With the equations provided, it is easy to build a distance map, for these equations to be used the C_X and C_Y parameters of the camera should be determined.

The angle of view of a camera can be determined as in Equation 4.1.10 below [49].

$$\alpha = 2 \arctan\left(\frac{l}{2f} \right) \qquad \textbf{(4.1.10)}$$

The parameter f is the focal distance of the lens; l is the length of the image sensor along the direction of the field of view and α is the field of view of the camera. The camera used in the system contains a 1/3'' CCD with R_X=720, R_Y=576 and lens with 2.8mm focal distance. To calculate the horizontal field of view, the physical dimension of the CCD along the horizontal direction l_H is needed, which can be found as in millimeters, knowing that the diagonal of the CCD is 1/3''.

$$l_H = \frac{R_X d}{\sqrt{R_Y^2 + R_Y^2}} = \frac{(720)\left(\frac{25.4}{3}\right)}{\sqrt{720^2 + 576^2}} = 6.6114mm \quad \textbf{(4.1.11)}$$

Similarly the length of the CCD along the vertical axis l_V is

$$l_V = \frac{R_Y d}{\sqrt{R_X^2 + R_Y^2}} = \frac{(576)\left(\frac{25.4}{3}\right)}{\sqrt{720^2 + 576^2}} = 5.2891mm \quad \textbf{(4.1.12)}$$

The field of view angles α_H and α_V in horizontal and vertical can then be computed using Equation 5.1.10.

$$\alpha_H = 2\arctan\left(\frac{l_H}{2f}\right) = 2\arctan\left(\frac{6.6114}{2(2.8)}\right) = 1.7361rad = 99.469° \quad \textbf{(4.1.13)}$$

$$\alpha_V = 2\arctan\left(\frac{l_V}{2f}\right) = 2\arctan\left(\frac{5.2891}{2(2.8)}\right) = 1.5137rad = 86.7291° \quad \textbf{(4.1.13)}$$

With the field of view angles, the C_X and C_Y parameters can be computed. Since the entire pixel array is covered in the field of view, the angle per pixel pitch can be determined in horizontal axis for C_X and vertical axis for C_Y as

$$C_X = \frac{\alpha_H}{R_X} = \frac{1.7361}{720} = 0.0024112rad / pixel = 0.1382° / pixel \quad \textbf{(4.1.14)}$$

$$C_Y = \frac{\alpha_V}{R_Y} = \frac{1.5137}{576} = 0.002628rad / pixel = 0.1506° / pixel \quad \textbf{(4.1.14)}$$

4.2 Determination of Laser Spot Coordinates

The video signal including the image of the laser beam is captured by the ADV7183 [28] video decoder IC on the DSP card. The video decoder provides ITU-R BT.656.4 compliant parallel data output relevant to the video frame data.

Figure 4.2.1 shows the composition of the data stream presented by ADV7183 to one of the Parallel Port Interfaces (PPI) of the BlackFin DSP.

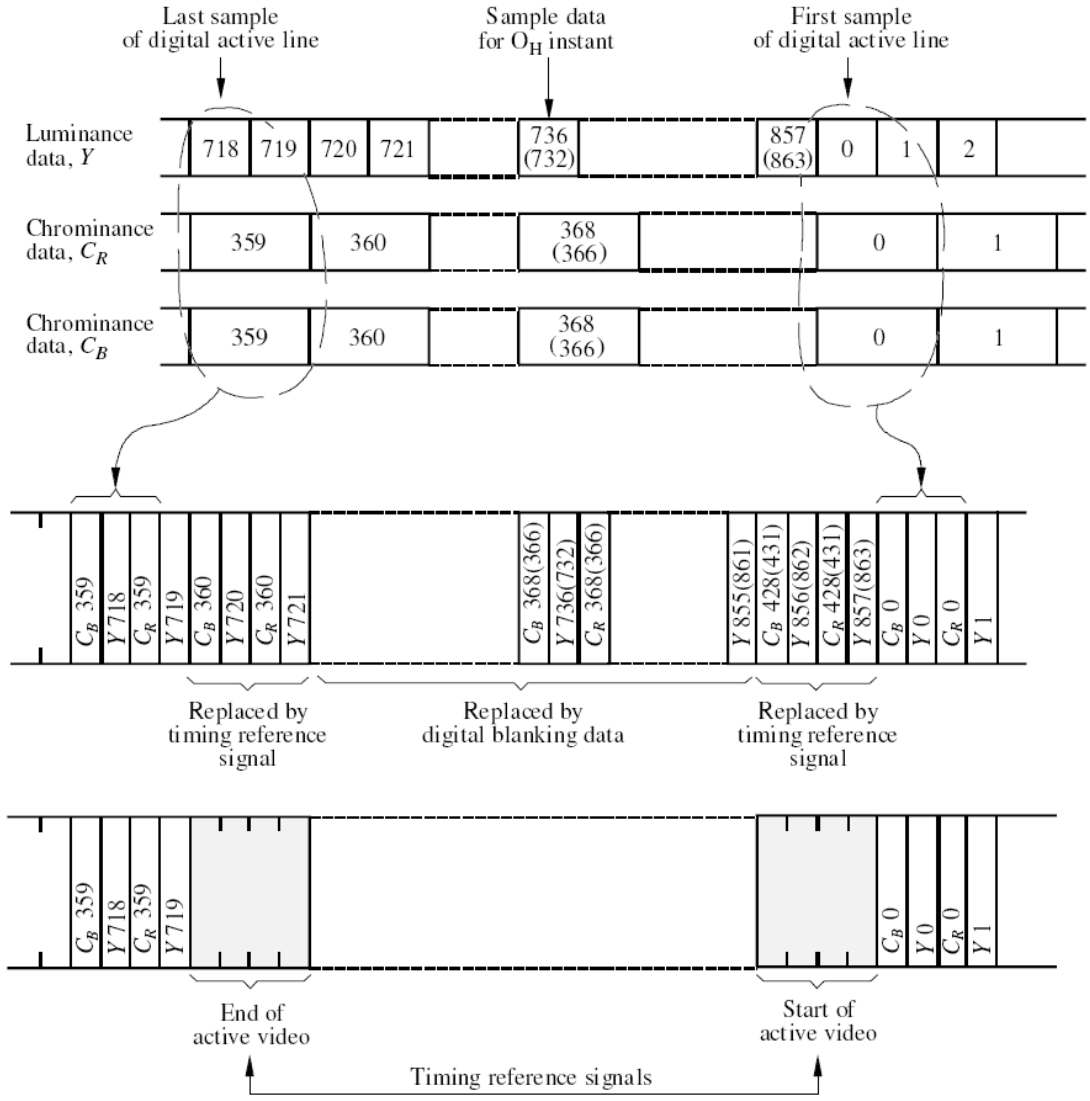

Figure 4.2.1 Composition of ITU-R 656 Data Stream

The data stream is copied to memory to a circular frame buffer consisting of four frames, by a Direct Memory Access (DMA) channel of the DSP. Each frame data in the memory includes timing reference signals, vertical blanking data and horizontal blanking data. In addition, according to the interlaced frame format the frame data consists of two consecutive fields, each containing either even or odd rows of the frame. Hence the frame data in the memory has the odd numbered row data first and the even numbered data next. All these frame format constraints emerging from the ITU format, makes it a requirement to apply a transformation to the coordinate values of a pixel of interest to find the actual place of the pixel in memory. In

addition, another transformation from luminance-chroma blue-chroma red (YCbCr) color definition to red-green-blue (RGB) color value is needed.

Assuming that a frame data is stored to an array of bytes named Frame, the Red (R), Green (G) and Blue (B) values of a pixel at (x, y) are accessed using the set of equations below [50]. The parameter B is defined for simplicity of equations.

$$B = \begin{cases} 288 + 4x + 1728\left(22 + \dfrac{y-1}{2}\right) & ,if\ y\ is\ odd \\ 288 + 4x + 1728\left(335 + \dfrac{y}{2}\right) & ,otherwise \end{cases} \quad \textbf{(4.2.1)}$$

$$Y = \frac{Frame[B+1] + Frame[B+3]}{2} \quad \textbf{(4.2.2)}$$

$$Cr = Frame[B+2] \quad \textbf{(4.2.3)}$$

$$Cb = Frame[B] \quad \textbf{(4.2.4)}$$

$$R(x,y) = 1.164(Y-16) + 1.596(Cr-128) \quad \textbf{(4.2.5)}$$

$$G(x,y) = 1.164(Y-16) - 0.813(Cr-128) - 0.392(Cb-128) \quad \textbf{(4.2.6)}$$

$$B(x,y) = 1.164(Y-16) + 2.017(Cb-128) \quad \textbf{(4.2.7)}$$

Once the transformation to access pixel values of the captured frames is set up, the next process defines a versatile way to position the laser spot accurately on the given video frame.

The most powerful clues in determining the position of the laser spot are the pure green color of the spot and the high intensity of the spot. To accurately locate the spot, both color and intensity properties should be exploited. However, the intensity of the laser spot shows wild changes with the reflectivity and color properties of target it reflects from. Hence, the major cue in finding the laser spot was selected as the color.

The reliability of the color of interest is very important for successful color segmentation. In many cases, this reliability can be maintained only within a

bounded volume in the RGB color space under fixed illumination conditions [51]. Namely, color segmentation suffers from the effect of illumination in many cases [24]. A number of color spaces exist in the literature, each one having its own advantages and disadvantages. For the design of a color segmentation algorithm, selection of the appropriate color space is one of the most important aspects [52]. Among these color spaces the Hue-Saturation-Intensity (HSI) color space is probably the most convenient in discriminating a color from others. In HSI color space, the hue component represents the dominant wavelength, saturation represents the purity of the color and intensity represent the amount of light [53], thus it is similar to the way that human beings understand and interpret color [54] and has a good performance in dealing with shadows, shades and highlights [52], because the hue component is insensitive to the illumination direction, intensity and the orientation of surfaces under white light [24].

The Hue (H), Saturation (S) and the Intensity (I) components are computed as follows [53].

$$H = \arctan\left(\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)}\right) \quad \textbf{(4.2.8)}$$

$$I = \frac{R+G+B}{3} \quad \textbf{(4.2.9)}$$

$$S = I - \frac{\min(R,G,B)}{I} \quad \textbf{(4.2.10)}$$

Segmentation algorithm was chosen to be one of the three methods described in [55]. Strong advantages of using HSI color system were stated above; however the hue coordinate is unstable in some degree. Small changes in RGB can cause strong variations in hue [56]. The system mainly suffers from three problems:

When the intensity average of the image is very low or extremely high, hue value is meaningless [55].

When the saturation is very low, hue does not contain significant information.

When the saturation is less than some minimum, the hue becomes unstable [55].

Fleyeh [55], describes three methods of HSI system color detection and segmentation, those can overcome the stated problems in some degree. The method used in this work is the first method described in his article, which takes saturation and illumination values into account to suppress the effects of known problems. The method basically depends on using the illumination mean of the given image to determine a threshold for color segmentation in hue – saturation plane.

After HSI conversion, the normalized intensity mean of the image which is in the range [0,1] is calculated as

$$\mu = \frac{1}{256.m.n} \cdot \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} I(x, y) \qquad \textbf{(4.2.11)}$$

Where m is size of the image in x axis, and n is the size of the image in y axis in pixels. I(x, y) is the intensity of the pixel at (x, y).

As stated earlier the hue and saturation are affected by the light conditions at which the image is taken, to some degree. Therefore the intensity mean calculated is used to calculate a threshold value as

$$\rho = e^{-\mu} \qquad \textbf{(4.2.12)}$$

This threshold is used to determine if a pixel can be considered as a predefined color or not, hence color segmentation of the image. The color that will be segmented is predefined by its average hue and saturation values. The reference color parameters and the parameters of a candidate pixel are represented by two vectors on hue – saturation plane. Figure 5.2.2 below summarizes the representation.

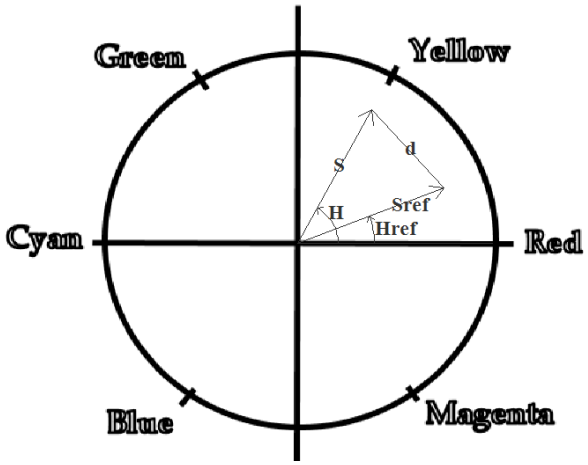

Figure 4.2.2 Definition of Euclidean Distance in H-S Plane

Where H_{ref} is the predefined hue value of the reference color, S_{ref} is the predefined saturation value of the reference color; H is the hue value of the pixel of interest, S is the saturation value of the pixel in interest and d is the Euclidean distance between two color vectors. The Euclidean distance between colors is calculated as follows:

$$d = \sqrt{\left(S.\cos H - S_{ref}.\cos H_{ref}\right)^2 + \left(S.\sin H - S_{ref}\sin H_{ref}\right)^2} \qquad \textbf{(4.2.13)}$$

The pixel is considered to be belonging to the reference color class if the distance is less than the threshold calculated with Equation 4.2.12, and it is considered as the background otherwise. The reference hue and saturation values for green were taken as 123 and 255 respectively.

After segmentation of the frame, regions close to pure green are obtained, and the region with the highest average intensity value is considered as the region created by laser. This assumption holds, since the system uses a 50mW laser, the brightness of the laser spot is extremely high even in bright daylight conditions.

There exists another challenge after finding the region related with the laser; finding the true place of the spot inside the region. This is a challenge, because the shape of the region is not always circular. To give an example of this kind of a situation, Figure 4.2.3 shows a magnified detail from a snapshot from the robot's camera where the green laser reflects from the target and scatters through the wall, approximately four meters to the robot camera.

Figure 4.2.3 Scattering of the Laser Beam

After doing the segmentation around the green color and finding the region in which the laser spot is, the only way to determine the real coordinates of the spot itself is to utilize the intensity. Figure 4.2.4 below shows the magnified intensity map and limits of the region in red of the occurrence shown in Figure 4.2.3 as it is found by the segmentation algorithm. The yellow cross denotes the actual position of the laser beam, found by the locating algorithm that will be described below

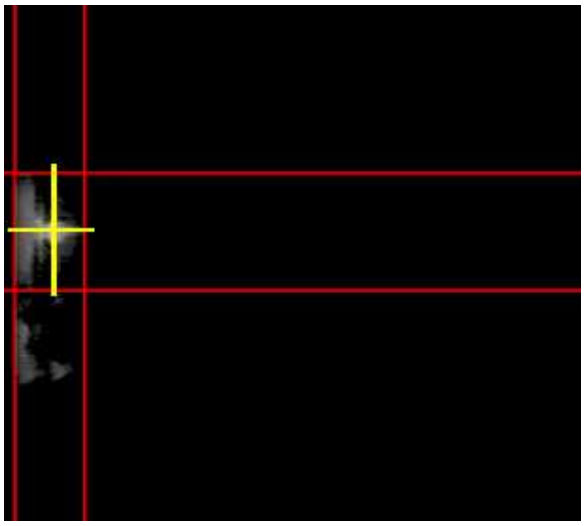

Figure 4.2.4 The Boundaries of the Problematic Region and the Process Result

To overcome the scattering problem and find the true location of the laser spot even in scattered situations, the first step is further thresholding in intensity inside the region. The mean of the intensity of the region is calculated similar to the process in Equation 4.2.11, but in this case only the intensities of the pixels that were

segmented as green were averaged. The threshold value for the intensity is chosen to be 75% of the maximum intensity level in the region, which was found empirically. This threshold removes a great portion of the scattered area. Then the center of intensity is found, similar to finding the center of mass in two dimensional objects, but using intensity values of pixels instead of weights of portions.

Let X_L, X_R, Y_U, Y_D, I(x, y) denote the left, right hand boundary x coordinates, upper, lower boundary y coordinates of the region and the intensity value of the pixel at (x, y). Then the total intensity mass m_T, the total intensity moment in horizontal direction M_X around left hand region boundary and the total intensity moment in vertical direction M_Y around upper region boundary can be found as

$$m_T = \sum_{y=Y_U}^{Y_D} \sum_{x=X_L}^{X_R} I(x, y) \quad \textbf{(4.2.14)}$$

$$M_X = \sum_{y=Y_U}^{Y_D} \sum_{x=X_L}^{X_R} I(x, y).(x - X_L) \quad \textbf{(4.2.15)}$$

$$M_Y = \sum_{y=Y_U}^{Y_D} \sum_{x=X_L}^{X_R} I(x, y).(y - Y_U) \quad \textbf{(4.2.16)}$$

After calculating total mass and moments of intensity, the center of intensity mass can be calculated easily. Since the center of mass should completely cancel the total moments in both directions when whole the intensity mass is thought to be at that point, the coordinates of the center of mass of intensity (G_X, G_Y) should be

$$(G_X, G_Y) = \left(\left(X_L + \frac{M_X}{m_T} \right), \left(Y_U + \frac{M_Y}{m_T} \right) \right) \quad \textbf{(4.2.17)}$$

The true location of the laser spot is always sufficiently close to (G_X, G_Y) including the instances with scattering problems.

5. REMOTE ACCESS CAPABILITIES

The importance of remotely accessing the robot has been discussed in the previous chapters. The robot utilizes a Wi-Fi connection on board, enabling the controllers to connect to the “Main Controller Board” of the system directly, and by having access to the “Main Controller Board”, the whole robot can be monitored and manipulated.

Since the main controller is a custom mini computer running Linux 2.6 kernel, and it has Ethernet connection; the remote debugging, controlling, file sharing, remote monitoring and all other concepts holding for two personal computers connected via network also hold for the robot and the host PC.

In addition, serial bootloaders written for LPC2138 microcontrollers, which are present on the two control cards for the camera position, LCD, base position and laser position controls; and VRS1000 microcontroller, which is present on the motor control circuits; make it possible to upload and test codes for these cards without requiring to deal with robot hardware. Moreover, since the DSP can be configured to boot from serial port, the DSP software also can be changed remotely by the operator.

The most useful remote access property in design phase was found to be ability to fetch the main controller card’s Linux console with Telnet. Some simple applications were written, for instance, for bridging the Telnet console to one of the serial ports, which was very useful in development of the initial versions of base controller; for updating the peripheral microcontrollers’ software, and for printing out the instantaneous system parameters on console which was very helpful before development of the PC control center.

Moreover, Ethernet connectivity cancels out a number of drudgery when developing applications for the main controller itself. The application to run on the main controller were written in PC, and compiled on Suse Linux by using arm-linux-gcc producing executables those are ready to work on the main controller. Then with a Telnet console, a NFS share of the PC Linux was mounted on the main controller.

After then the compiled programs, data related to these programs, required directory structure and similar things were formed on the shared directory of the PC, and virtually on the main controller complete tests, as if the software in development was in the original ramdisk of the main controller could be performed.

To demonstrate some of these networking induced properties, and to test the robot mechanics, electronics and software which had been in continuous development for a long period, a PC control platform working on Windows operating system was written. Figure 5.1 below shows a snapshot of the main form of PC control platform in operation.

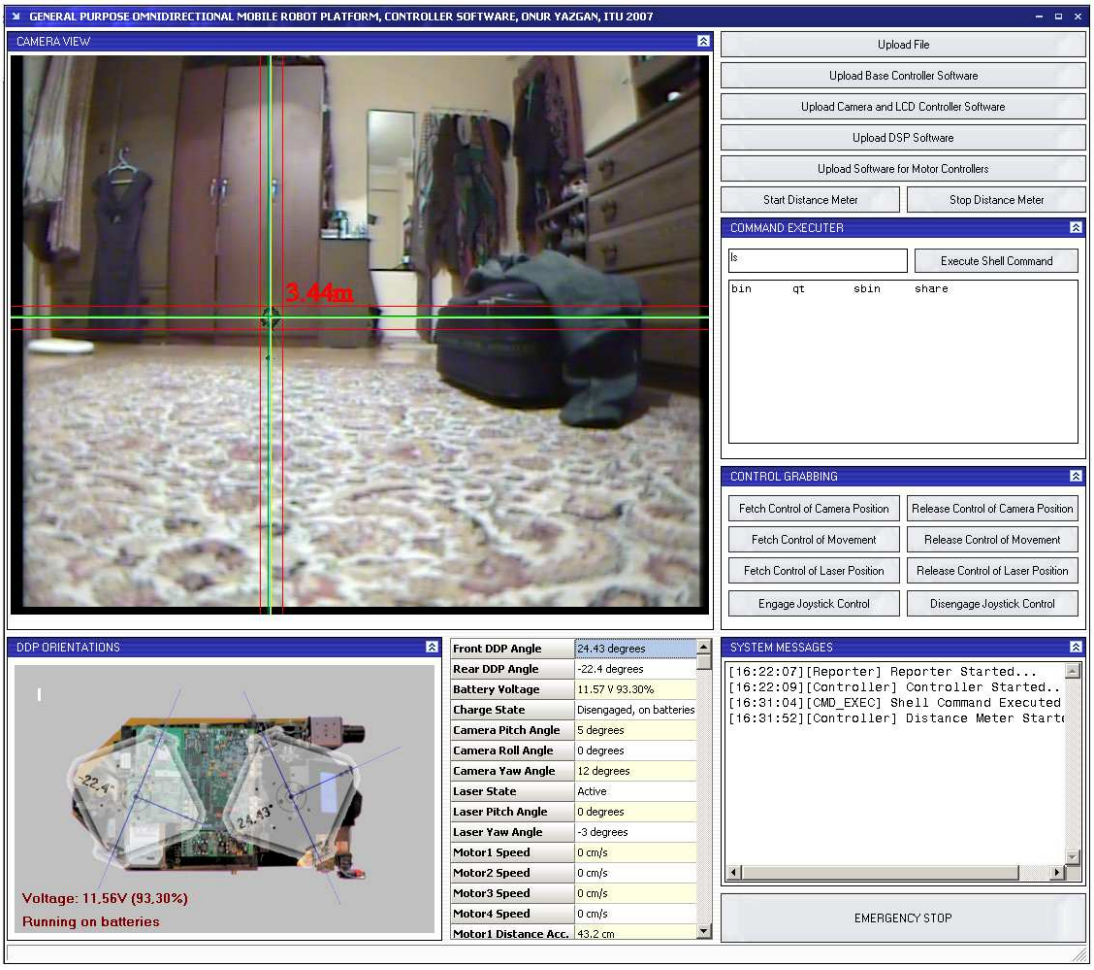

Figure 5.1 Main Screen of the Remote Control Platform

The remote control platform is in connection with various client programs which are running on the robot via UDP protocol. The whole parameter set of the robot is indicated in real time, while the angles of the DDPs are also shown graphically on an animated representation. The processed image from the robot's DSP system is

transferred to the demodulator and frame grabber components which are attached to PC externally. The video stream is then captured and shown on the main form. The only lag between the actual DSP video stream output and the video observed on the PC screen is the lag resulting from the stream buffers of both the frame grabber system and the video-for-windows library.

The remote control platform is capable of executing shell commands on main controller and getting responses, uploading any kind of file to device as well as uploading and updating the software of “Camera Control and LCD Bridge Card”, “Laser and Base Controller Card” and of the “Motor Controller Cards”. The software is also capable of capturing debug messages and gathering control of each system resource at any time.

Moreover, to evaluate the mechanical structure and the controllers’ performance a joystick link has been set up. The joystick can be used for moving the robot, moving the camera, moving the laser and controlling the laser emission.

6. CONCLUSION

The main purpose of this thesis was to develop a mobile robot platform suitable for development of applications concerning any kind of theory related to robotics and control. As it was clarified during the text, this simple definition of the required system brought many hard-to-achieve properties that the final design would have. Summing up some of these dictated properties, the final design can be called "An Omnidirectional Mobile Robot Platform with Broadband Data Connection, High Resolution Image and Audio Processing Capabilities, Three-Dimensional Environment Mapping Capability and Internal Cooperation of Mechanics and Software that Reduces Odometric Errors". Nevertheless, a design solution satisfying all needed key concepts was achieved and a prototype was built.

During the design and implementation of the robot, the most challenging issue was the mechanics. Although the idea of using two differential drive platforms to form a four-degree-of freedom omnidirectional base was present before starting the actual building phase of the mechanics, the details of other systems brought the process to a stuck situation twice, such that no solution for a problem on a detail could be found. In the first attempt shown in Figure 6.1, the mechanism which was built using stepper motors and synchronous belt to move the camera base could not be operated.

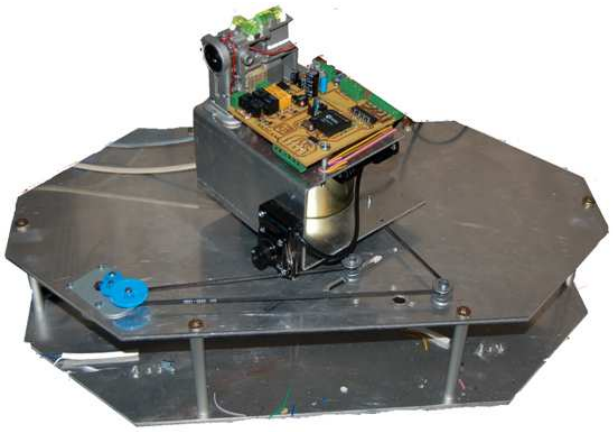

Figure 6.1 The First Attempt of Building the Robot

To avoid the problem faced in the first attempt, the camera base was designed to rotate around the shaft of a motor. The motor was fixed to the camera base and the shaft was fixed to the upper platform as they can be seen in Figure 6.2. At the end of this attempt the cables of camera, the camera motor control system and the laser were crushed between the rotating camera base and the upper platform.

Figure 6.2 The Second Attempt of Building the Robot

In both cases of mechanical failures the whole mechanic details were redesigned in each situation and new mechanics were built up. The solution presented in this context and shown to hold the design criteria of the concept was the third robot that has been built which is shown in Figure 6.3.

Figure 6.3 The Final Design

Dealing with the possible improvements that may be achieved, the most significant ones are about the odometric error reduction mechanism and the laser range measurement system.

As stated in Section 3.5.2, a further improvement about the capability of odometry error compensation of this approach has already been proposed by Borenstein [10]. Borenstein adds a sliding mechanism to one of the DDPs, which makes the DDP move freely about the principal axis of the robot. The length of this link, which he calls a compliant-linkage, is measured. In the static link design proposed in this thesis, the position of the DDP that has not induced an error is updated to match the length between DDPs and hence limiting accumulation of errors by Equations 3.5.2.16 and 3.5.2.17, where we take the link length L as a constant. However, Borenstein [21] states that the momentary controller errors would make the DDPs move closer or far from each other and the function of the compliant linkage is letting DDPs move freely to make errors which is inevitable, then to measure the link length and correct the estimated positions of the DDPs. Since it is a well known fact that controller errors are inevitable, letting errors express themselves without distorting other components of the system and then fixing the error with accurately measured references is a very logical way.

A strong negative effect of fixing the link length in the design presented in this thesis was not observed. The most plausible explanation for this situation is based on the fact that the drive motors used in DDPs are so power full that, even a single DDP could move the whole robot. To clarify the reason, assume that the controller errors of DDPs dictate the back DDP motors to turn slower than the front DDP such that if there was a compliant linkage between them, the back DDP would lag and the length of the link would grow. However, in constant link design, the back DDP is mechanically attached to the front DDP, concluding that as it is not allowed lag, it will exert a force on the front DDP to the direction of lag; and as the front DDP is strong enough to take this extra load easily, it will basically drag the back DDP along with matching the speed of the back DDP to its speed. Conversely, if the controller errors result in a way that the back DDP motors turn faster than the front, which would result the back DDP to get closer to the front in compliant linkage design, the back DDP will push the front easily to match its speed.

A significant improvement on the laser range measurement system can be achieved if liner type lasers are used rather than a point type laser. The liner lasers provide a straight line of laser instead of a point. By using this kind of a laser a whole cross-section of the environment in the field of view of the camera can be constructed from a single frame. This will also cancel the yaw motor of laser which is used to move the laser beam on a line to get a cross-section.

The advantages of using a liner type laser were realized earlier in the conceptual design phase of the system. Besides, a liner type red laser was used in the first two design attempts as it can be realized from Figures 6.1 and 6.2. The photograph in Figure 4.1 was taken using that liner laser. The improvement that can be achieved by a liner laser is an obvious fact, however, the deficiencies of the laser and the camera system limits usage of liners.

As it was shown in Section 4.2, the color and the brightness information to determine the regions of laser is important in color segmentation. The red line laser used in trials was a simple 630nm red laser with 1mW peak output power. Since the laser beam is spread into a wide line, the laser power, hence the brightness of each point on the line is reduced by a factor that can be approximated as the ratio of the area of a single laser point occupies at the distance of interest over the total length of the laser line, which is a quite effective reducing factor. Moreover, the general spectral sensitivity of CCD cameras for red light is lower when compared to blue and green lights. Resulting that, especially in highly illuminated conditions like daylight, the laser line in the captured images was generally indistinguishable even by a human eye. To overcome this problem, a search for high power liner type laser modules was performed and it was determined that there are a very few models available at the order of 5mW in terms of power, which were quite expensive. As a consequence, the way of building a mechanical scanning system with a high power point laser was selected. It is obvious that it would present a great improvement on system performance if a liner type; high power laser could be utilized.

Another improvement on the laser range measurement system can be simply using a camera with higher resolution, which would enable to make more precise and increase the available range of measured distances. Moreover, it can be a further improvement if the current CCD is camera is not replaced, instead a second camera

with higher resolution is added. This would enable applying stereo vision algorithm to provide further information about the environment and a possibility of fusing the data from the stereo vision with the laser range measurements to provide more detailed and accurate environment models.

To sum up; an omnidirectional mobile robot design and implementation was completed in the scope of this thesis. The mechanical design of the robot shows great advantages by means of odometric accuracy. Moreover, the overall design satisfies all key concepts that were summarized as moving flexibility, speed and accuracy; sufficient internal and external sensing; sufficient and flexible processing power and sufficient communication facilities to match the main design objective of being a base design platform for application development in researches of robotics.

REFERENCES

[1] **Gray, J. O., & Caldwell, D. G.,** 1996. *Advanced Robotics & Intelligent Machines.* London: The Institution of Electrical Engineers.

[2] **Wiklund et al.,** 1988. AGV Navigation by Angle Measurement, *Proceedings of the 6th International Conference on Automated Guided Vehicle Systems 1988,* Brussels, (pp. 199-212).

[3] **Pritschow, G., Jantzer, M., & Schumacher, H.,** 1988. A Control System for Free-Ranging Mobile Robots in Production Lines. *Proceedings of the 6th International Conference on Automated Guided Vehicle Systems 1988*, Brussels, (pp. 189-197).

[4] **Borenstein, J., & Koren, Y.,** 1985. A Mobile Platform for Nursing Robots, *IEEE Transactions on Industrial Electronics*, ***32*** (2), 158-165.

[5] **Borenstein, J.,** 1993. Multi-Layered Control of a Four-Degree-of-Freedom Mobile Robot with Compliant Linkage, *Proceedings of the 1993 IEEE International Conference on Robotics and Automation,* Atlanta, (pp. 3.7-3.12).

[6] **Borenstein, J.,** 1992. Compliant-Linkage Kinematic Design for Multi-Degree-Of-Freedom Mobile Robots, *SPIE Symposium on Advances in Intelligent Systems,* Boston, (pp. 344-351).

[7] **Leifer, L. J., Van der Loos, H. F., & Chalowski, S. J.,** 1988. Development of an Omnidirectional Mobile Vocational Assistant Robot, *Proceedings of the International Conference of the Association for the Advancement of Rehabilitation Technology,* Montreal.

[8] **Feng, D., Friedman, M. B., & Krogh, B. H.,** 1989. The Servo-Control System for an Omnidirectional Mobile Robot, *Proceedings of the 1989 IEEE*

International Conference on Robotics and Automation, Arizona, (pp. 1566-1571).

[9] **Killough, S. M., & Pin, F. G.,** 1992. Design of an Omnidirectional Holonomic Wheeled Platform Prototype. *Proceedings of the IEEE Conference on Robotics and Automation,* Nice, (pp. 84-90).

[10] **Borenstein, J., & Evans, J.,** 1997. The OmniMate Mobile Robot-Design, Implementation, and Experimental Results, *Proceedings of the IEEE International Conference on Robotics and Automation,* Albuquerque, (pp. 3505-3510).

[11] **Reister, D. B.,** 1991. A New Wheel Control System for the Omnidirectional HERMIES-III Robot, *Proceedings of the IEEE Conference on Robotics and Automation,* Sacramento, (pp. 2322-2327).

[12] **Moravec, H. P.,** 1984. *Three Degrees for a Mobile Robot,* Carnegie-Mellon University: The Robotics Institute, Mobile Robots Laboratory.

[13] **West, M., & Asada, H.,** 1992. Design of a Holonomic Omnidirectional Vehicle, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation,* Nice, (pp. 97-103).

[14] **Hirose, S., & Amano, S.,** 1993. The VUTON: High Payload Efficiency Holonomic Omni-Directional Vehicle, *Proceeding of the 6th International Symposium on Robotics Research,* Hidden Valley.

[15] **Pin, F. G., & Killough, M.,** 1994. A New Family of Omnidirectional and Holonomic Wheeled Platforms for Mobile Robots, *IEEE Transactions on Robotics and Automation*, ***10*** (4), 480-489.

[16] **Chenavier, F., & Crowley, J.,** 1992. Position Estimation for a Mobile Robot Using Vision and Odometry, *Proceedings of IEEE International Conference on Robotics and Automation,* Nice, (pp. 2588-2593).

[17] **Evans, J. M.,** 1994. HelpMate: An Autonomous Mobile Robot Courier for Hospitals, *1994 International Conference on Intelligent Robots and Systems,* Munich, (pp. 1695-1700).

[18] **Borenstein, J., & Feng, L.,** 1996. Measurement and Correction of Systematic Odometry Errors in Mobile Robots. ***12*** (6).

[19] **Borenstein, J., Feng, L., & Wehe, D.,** Mobile Robot Positioning - Sensors and Techniques. *Journal of Robotic Systems, Special Issue on Mobile Robots*, ***14*** (4), 231-249.

[20] **Byrne, R. H., Klarer, P. R., & Pletta, J. B,** 1992. *Techniques for Autonomous Navigation,* Albuquerque: Sandia Report SAND92-0457, Sandia National Laboratories.

[21] **Borenstein, J.,** 1995. Internal Correction of Dead-Recogning Errors with a Dual-Drive Compliant Linkage Mobile Robot, *Journal of Robotic Systems*, ***12*** (4), 257-273.

[22] **Cox, I. J.,** 1991. Blanche - An Experiment in Guidance and Navigation of an Autonomous Mobile Robot. *IEEE Transactions Robotics and Automation*, ***7*** (2), 193-204.

[23] **Pugh, A.,** 1983. *Robotic Technology.* London: Peter Peregrinus Ltd.

[24] **Kim, C. H., You, B. J., & Kim, H.,** 2006. Color Segmentation Robust to Brightness Variations by using B-Spline Cure Modeling. *IEEE International Conference on Intelligent Robots and Systems.* Beijing.

[25] **Gordon, G., Darell, T., Harville, M., & Woodfill, J.,** 1999. Background Estimation and Removal Based on Range and Color, *IEEE Workshop on Visual Surveillance,* Bombay.

[26] **Cirrus Logic Inc.,** EP9307 Data Sheet. (2005, May). Austin, USA.

[27] **Analog Devices Inc.,** ADSP-BF561 EZ-KIT Lite Evaluation System Manual. (2007, January). Norwood, USA.

[28] **Analog Devices Inc.,** ADV7183A Multiformat SDTV Video Decoder Datasheet. (2005). Norwood, USA.

[29] **International Telecommunication Union,** (1986-1992-1994-1995-1998), Recommendation ITU-R BT.656-4.

[30] **Analog Devices Inc.,** ADV7179 Chip Scale PAL0NTSC Video Encoder with Advanced Power Management Datasheet. (2004). Norwood, USA.

[31] **Behnke, S., & Schreiber, M.,** 2006. Digital Position Control for Analog Servos, *Proceedings of the Workshop on Humanoid Soccer Robots of*

the 2006 IEEE-RAS International Conference on Humanoid Robots, Genoa.

[32] **Goal Semiconductor.**, VERSA 1000: 8-Bit, 40MHz, 1K RAM and 64K Embedded ISP FLASH MCU Datasheet. (2002). Montreal, Canada.

[33] **ST Microelectronics.,** L298 Dual Full-Bridge Driver Datasheet. (2000, January). Geneva, Switzerland.

[34] **Atmel Corporation.,** AVR221: Discrete PID Controller. (2006). California, USA.

[35] **Chen, C. T.,** 1993. *Analog and Digital Control System Design: Transfer Function, State-Space, and Algebraic Methods.* Fort Worth: Saunders College Publishing.

[36] **Johnson, M. A., & Moradi, M. H.,** 2005. *PID Control: New Identification and Design Methods.* London: Springer-Verlag Limited.

[37] **Goodwin, G. C., Graebe, S. F., & Salgado, M. E.,** 2001. *Control System Design.* New Jersey: Prentice Hall.

[38] **Ziegler, J. G., & Nichols, N. B.,** 1942. Optimum Settings for Automatic Controller. *Trans.ASME*, *64*, 759-768.

[39] **Philips Semiconductors.,** LPC 2138 Single-chip 16/32-bit Microcontrollers; 512kB ISP/IAP Flash with 10-bit ADC and DAC Datasheet. (2005, April 15). Eindhoven, Netherlands.

[40] **National Semiconductor.,** LM1117/LM1117I 800mA Low-Dropout Linear Regulator Datasheet. (2002, October). California, USA.

[41] **Microchip Technology Inc.,** MCP6001/2/4 1MHz, Low-Power Op Amp Datasheet. (2005). Arizona, USA.

[42] **National Semiconductor.,** LM4041 Precision Micropower Shunt Voltage Reference Datasheet. (2005, March). California, USA.

[43] **Borenstein, J., & Feng, L.,** 1995. Correction of Systematic Odometry Errors in Mobile Robots, *Proceedings of the 1995 International Conference on Intelligent Robots and Systems,* Pittsburgh, (pp. 569-574).

[44] **Kurazume, R., & Nagata, S.,** 1994. Cooperative Positioning With Multiple Robots, *Proceedings of IEEE International Conference on Robotics and Automation,* Nice, (pp. 1250-1257).

[45] **Banta, L.,** 1988. A Self Tuning Navigation Algorithm. *Proceedings of the 1988 IEEE International Conference on Robotics and Automation,* Philadelphia, (pp. 1313-1314).

[46] **Fu, K. S., Gonzales, R. C., & Lee, C. S.,** 1987. *Robotics: Control, Sensing, Vision and Intelligence.* New York: McGraw-Hill.

[47] *Details of the Laser Range Finder*. (n.d.). Retrieved 11 13, 2005, from UB Robotics: http://www.eng.buffalo.edu/ubr/ff03laser.php

[48] *Webcam Based DIY Laser Rangefinder*. (n.d.). Retrieved 11 13, 2005, from http://www.pages.drexel.edu/~twd25/webcam_laser_ranger.htm

[49] *Angle of View*. (2007, December 7). Retrieved December 10, 2007, from Wikipedia: http://en.wikipedia.org/wiki/Angle_of_view

[50] **Jack, K.,** 1997. YCbCr to RGB Considerations. California, USA: Intersil Americas Inc.

[51] **Cho, K., Jang, J., & Hong, K.,** 2001. Adaptive Skin Color Filter. *Pattern Recognition , 34*, 1067-1073.

[52] **Tsang, P. W., & Tsang, W. H.,** 1996. Edge Detection on Object Color. *IEEE International Conference on Image Processing*, (pp. 1049-1052).

[53] **Azziz, M. Z., Shafik, M. S., Mertsching, B., & Munir, A.,** 2005. Color Segmentation for Visual Attention of Mobile Robots. *IEEE 2005 International Conference on Emerging Technologies,* Islamabad, (pp. 115-120).

[54] **Carron, T., & Lambert, P.,** 1994. Color Edge Detector Using Jointly Hue, Saturation and Intensity. *IEEE Conference on Image Processing,* Austin, (pp. 977-1008).

[55] **Fleyeh, H.,** 2004. Color Detection and Segmentation for Road and Traffic Signs. *IEEE Conference on Cybernetics and Intelligent Systems,* Singapore, (pp. 809-814).

[56] **Lalonde, M. L.,** 1995. *Road Sign Recognition, Survey of the State of the Art.* Montreal: Centre de Recherche Informatique de Montreal.

RESUME

Onur YAZGAN was born in İzmir, in 1981. He completed his higher education at İzmir 60th Year Anatolian High School, in June 1999. He received his Bachelor of Science degree from Istanbul Technical University, Electronics and Communication Engineering Department in 2004. Same year, he was accepted to Istanbul Technical University Computer Engineering Department for Master of Science in Engineering programme. He has been working as a hardware and software design engineer for Adam ArGe Telecommunications Research and Development Ltd., since 2002.