

# A Novel Positioning System for accurate tracking in Indoor Environments

by

Srujan Linga

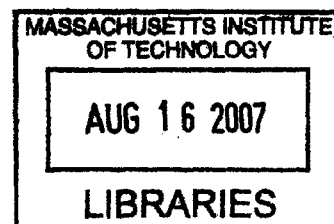
Submitted to the Department of Electrical Engineering and Computer Science  
in partial fulfillment of the requirements for the degree of  
Master of Science

ARCHIVES

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007



© Massachusetts Institute of Technology 2007. All rights reserved.

Author.....  
Department of electrical Engineering and Computer Science  
June 2007

Certified by.....  
Daniela Rus  
Professor  
Thesis Supervisor

Accepted by.....  
Arthur C. Smith  
Chairman, Department Committee on Graduate Students



# **A Novel Positioning System for Accurate Tracking in Indoor Environments**

by

Srujan Linga

Submitted to the Department of Electrical Engineering and Computer Science  
in May 2007, for the partial fulfillment of the  
requirements for the degree of  
Master of Science

## **Abstract**

Precise positioning is crucial to many applications involving autonomous robots in indoor environments. Current solutions to the indoor localization problem are either both highly unreliable and inaccurate (like GPS based systems), or hugely expensive (such as the iGPS system). In this thesis we propose, design and build a low-cost, robust and highly accurate indoor localization system using laser light sources. The system is composed of three transmitting laser modules arranged in a straight line and a receiver module mounted on the mobile robot. The system calculates the coordinates of the mobile robot by using triangulation algorithms which require precisely measured values of the angles of the receiver with respect to the three laser emitters. Results from practical testing of the system in an aircraft wing assembly set-up have been found to be extremely encouraging. Using our system, the mobile robotic arm could be localized accurately within error margins defined approximately by Gaussian distributions centered at the object's true coordinate values and with standard deviations of 0.1778 mm, 0.1016 mm and 0.3352 mm in the  $x$ ,  $y$  and  $z$  coordinate directions respectively. The system is also used to detect height drop in the arm due to its weight as it extends to perform fitting operations on the skin of the wing. Feedback from the laser localization system is used to adjust the position of the tip of the robotic arm in order to perform a sequence of high precision docking tasks within the aircraft wing.

Thesis Supervisor: Daniela Rus

Title: Professor

This page has been intentionally left blank

# Acknowledgements

This research would not have been possible without the continual guidance, support and supervision of my advisor Daniela Rus. I would also like to thank all the members of the Distributed Robotics Lab at MIT. In particular, I would like to thank Keith Kotay who helped me greatly with the design of the system hardware and Iuliu Vasilescu for his insightful discussions and help in software programming. The aircraft wing assembly experiments could not have happened without Binayak Roy and Manas Menon who helped tremendously in deploying the system and collecting data. Last but not the least; I would like to thank my family and friends to whom I am grateful for their support and encouragement.

This page has been intentionally left blank

# Table of contents

<b>1 Introduction</b> .....	<b>11</b>
1.1 Motivation .....	11
1.1.1 Triangulation.....	12
1.1.2 Scene Analysis.....	13
1.1.3 Proximity.....	13
1.1 Contributions & Thesis Outline .....	14
<b>2 Related Work</b> .....	<b>17</b>
2.1 Background .....	17
2.1.1 Odometry.....	18
2.1.2 Active Beacon .....	18
2.2 A survey of locationing systems .....	19
2.2.1 Active badges.....	19
2.2.2 Active Bats .....	20
2.2.3 Cricket.....	21
2.2.4 Radar.....	22
2.2.5 Electromagnetic sensors.....	23
2.2.6 Computer Vision techniques .....	24
2.2.7 Smart environments .....	25
2.2.8 Sensor Fusion.....	25
2.2.9 iGPS System .....	26
<b>3 Algorithms</b> .....	<b>29</b>
3.1 Problem Statement .....	29
3.2 Solution .....	30
3.2.1 Transmitter placement in a given 3D space .....	31
3.3 Transmitter Identification.....	33
3.4 Solution to the localization problem .....	34
3.5 Localization technique .....	35

**This page has been intentionally left blank**



<b>4 Hardware platform</b> .....	<b>41</b>
4.1 Hardware components.....	41
4.1.1 The laser module .....	41
4.1.2 Photodiode sensor module .....	43
4.1.3 Central control and processing unit .....	46
<b>5 System Design</b> .....	<b>49</b>
5.1 System overview .....	49
5.2 Angle computation .....	50
5.3 State machine design.....	53
<b>6 Experiments and Results</b> .....	<b>55</b>
6.1 Motivation & experimental set-up .....	55
6.2 Experimental Results.....	60
6.2.1 Angular accuracy and distributions .....	60
6.3 Theoretical error analysis .....	64
6.4 Real time tracking .....	67
6.5 Cross Validation.....	69
<b>7 Conclusions</b> .....	<b>74</b>
7.1 Lessons learned .....	75
7.1.1 Sensitivities .....	77
7.1.2 Limitations .....	78
7.2 Future Work .....	79
<b>Appendix A</b> .....	<b>81</b>
<b>Appendix B</b> .....	<b>83</b>
B-1. FPGA code for central control and processing unit .....	83
B-2. Java code for reading data from CCPU to computer .....	87
B-3. MATLAB code for coordinate calculation and 3D plotting.....	89
<b>Bibliography</b> .....	<b>91</b>

This page has been intentionally left blank

# Chapter 1

## Introduction

In this chapter we present a general overview of the localization problem and why it is important in today's world. In addition, we also discuss the most important techniques currently used to solve the localization problem.

### 1.1 Motivation

There is an increasing number of potential applications for autonomous mobile robots in indoor environments, ranging from cleaning, to surveillance, to search and rescue operations in burning buildings or hostage situations, to assisting the handicapped or elderly around the home. During the last few decades it could also be observed that robots have increasingly and successfully spread in to almost every field of manufacturing and production in order to assist or even completely replace the human operator to perform difficult, tedious or dangerous jobs. Popular examples range from small scale applications such as wire bonding in chip manufacturing or mounting the mechanical parts of a wristwatch up to large scale tasks such as welding a car body, transportation of parts in a plant by automated guided vehicles or assembly operations on the International Space Station. In order for the successful use of mobile robots in all of the above applications, two crucial technical challenges that must be dealt with are i) the ability of the robot to self-localize and ii) building a robust way of ensuring that the robot can follow a particular path given the unavoidable odometric and control errors that must be dealt with for any moving robot.

The central function of a self-localization system is to provide the robot with accurate positional data. One can think of different ways of expressing the information

about a robot's position in a given environment. It could be in relation to a global coordinate system or it could be relative to some object in its neighborhood. In most systems, a combination of both the above methods is needed as every physical contact requires the robot to position itself relative to the object while it also needs the global coordinates in order to compute an optimal path to travel from one point to another. Traditional odometric systems [1] relied on the knowledge of history of movements in order to calculate the current position and thus failed to work in the absence of step-by-step information about how the current position of the robot was achieved. This is the same problem as initializing the position of a robot when it is first powered up. Thus, traditionally, most robots have shown position tracking capabilities and relied on manual initialization which is inadequate for their complete autonomy.

When attempting to determine the instantaneous location of a mobile robot without the knowledge of historical data, one can choose from three major techniques:

### **1.1.1 Triangulation**

Triangulation uses the geometric properties of triangles to compute locations. Triangulation can be divided into the sub-categories of lateration, using distance measurements, and angulation, using primary angle or bearing measurements. Lateration computes the position of an object by measuring its distance from multiple reference points. Calculation of an object's position in 2D requires distance measurements from 3 non-collinear points as shown in Fig. 1-1. In 3D, distance measurements from 4 non-collinear points are required. Domain specific knowledge may reduce the number of required distance measurements [2]. Distance measurements can be obtained by three general approaches: i) Direct measurement using physical action or movement (eg. A robot extending an arm till it touches a surface), ii) Time-of-Flight measurement (eg. using sound waves to measure the distance) and iii) Attenuation of intensity of the emitted signal (eg. free space radio waves approximately attenuate in intensity proportional to the squared distance).

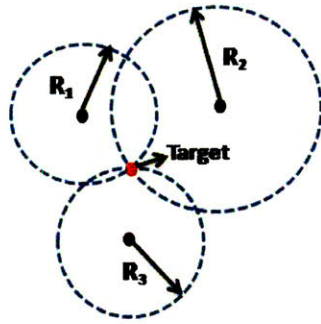


Figure 1-1: Lateralation

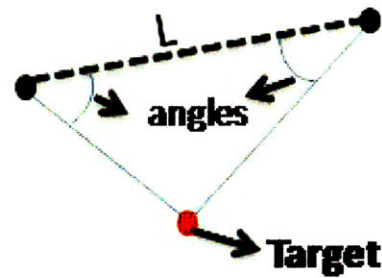


Figure 1-2: Angulation

On the other hand angulation uses angles in order to determine the position of an object. In general, two dimensional angulation requires two angle measurements and one length measurement between two reference points as shown in Fig. 1-2. In 3D, one length measurement, one azimuth measurement and two angle measurements are needed to specify a precise location (eg., phased antenna arrays).

### 1.1.2 Scene Analysis

The scene analysis location sensing technique use features of a scene observed from a particular vantage point to draw conclusions about the location of the observer or of objects in the scene. Usually the observed scenes are simplified to obtain features that are easy to represent and compare (e.g., the shape of horizon silhouettes in [3]). In static scene analysis, observed scenes are looked up in a predefined table that maps them to object location. In contrast, differential scene analysis tracks the difference between successive scenes to estimate location.

### 1.1.3 Proximity

Proximity location sensing technique entails determining when an object is “near” a location. The object presence is sensed using physical phenomena with limited range. There are three general approaches to sensing proximity: i) Detecting physical contact (e.g., using pressure sensors, touch sensors, capacitive field detectors, etc), ii) Monitoring

wireless cellular access points, and iii) Observing automatic ID systems (e.g, using RFID tags, UPC barcodes, etc.).

## **1.1 Contributions & Thesis Outline**

In order to realize the full potential of autonomous robots whether they are being used for domestic applications (e.g., assisting the elderly, search and rescue teams, etc) or to perform large-scale assembly tasks, accurate positioning is one of the most crucial problems to solve. Automation of large-scale assembly tasks such as assembly of terrestrial buildings, planetary habitats, airplane wings, space solar power structures, etc, has long been a challenging problem to solve in the manufacturing/construction industry. In order to achieve complete automation of large-scale assembly not only does one require robust coordination of heterogeneous robots but also high precision locationing of assembly tools and robots to perform accurate docking tasks between individual structures. In assembly operations, the fact that one requires locationing to be accurate in the order of millimeters over a large area of operation is what makes the problem difficult and challenging to solve. In the current work we address this problem by developing a novel device and algorithms to achieve highly accurate localization (in the order of sub-millimeters) and evaluate its performance in an airplane wing assembly set-up. Traditional solutions to the localization problem either involve relative position measurement systems such as odometric systems, inertial navigation systems, etc or absolute position measurement systems such as magnetic compasses, landmark navigation systems, GPS, etc [4]. As explained in chapter 2, the state-of-art in each of these systems can at best achieve precision up to a few centimeters [4]. GPS is an excellent technology for outdoor navigation which works based on the times of travel of RF signals emitted by satellites. However, GPS is imprecise with inaccuracies up to 100m maximum due to intentional degradation of accuracy for security purposes [4]. Moreover, the reception of GPS signals inside most buildings is not reliable. Hence, in many applications, such as manufacturing automation where precise indoor locationing is needed, GPS based systems fail. Other solutions to the locationing problem include the

use of camera based monitoring systems which use CCD camera sensors [5] to monitor the robot's movements and laser based systems such as LIDAR (*Light-Imaging Detection and Ranging*) which use the properties of scattered laser pulses to find the range of a distant target. However, both the systems are highly expensive and do not achieve accuracy levels in the order of sub-millimeters. Thus, the main contributions of the current thesis are as follows:

- i) We propose, design and build a low-cost laser based localization system that achieves sub-millimeter accuracy levels,
- ii) Present algorithms for laser-based localization,
- iii) Experimentally evaluate and analyze the system in an airplane wing assembly set-up to track the position of a robotic arm that performs difficult fitting operations.

The remainder of the thesis is organized as follows: chapter 2 consists of literature review of the state-of-the-art in indoor locationing. Chapter 3 discusses problem formulation and algorithms for localization using lasers followed by a description of the hardware in chapter 4. Chapter 5 focuses on system design and computational techniques. Experimental results of using the system in airplane wing assembly are presented in chapter 6, followed by the conclusions in chapter 7.

Note: All the hardware and software files used in the implementation of the localization system presented in this thesis can be accessed online at: <http://groups.csail.mit.edu/drl/wiki/index.php/lasersystem>

This page has been intentionally left blank



# Chapter 2

## Related Work

In this chapter, we explain in detail the most popular commercial and research systems currently in use for localization of mobile robots and also evaluate them in terms of their localization accuracy, cost and other important engineering criteria.

### 2.1 Background

Before we present a literature survey of approaches that deal with localization in indoor environments, it is important to understand that robot localization task cannot be viewed independently. Localization provides a positional fix of the robot in its environment. However, this information is actually useful if a reference frame is given, i.e., if the positional fix is specified with respect to a geometrical or topological map of the surroundings. In cases where such a map is not priori given the robot has to perform concurrent localization and map building [6]. The focus of the current work is on scenarios where the environmental map is priori given and one has to calculate the precise coordinates of the robot relative to it. In the current chapter, after describing the major techniques currently used for localization, a detailed description and comparison of various commercially available instruments will be presented. The most prominent techniques used for localization are:

### **2.1.1 Odometry**

Odometry [1] is one of the most widely used navigation methods for mobile robot positioning; it provides good short-term accuracy, is inexpensive, and allows very high sampling rates. However, the fundamental idea of odometry is the integration of incremental motion information over time, which leads inevitably to the unbounded accumulation of errors. Specifically, orientation errors will cause large lateral position errors which increase proportionally with the distance traveled by the robot. Odometry is based on simple equations, which hold true when wheel revolutions can be translated accurately into linear displacement relative to the floor. However, in the case of wheel slippage and some other more subtle causes, wheel rotations may not translate proportionally into linear motion. The resulting errors can be categorized into one of two groups: systematic errors and non-systematic errors. Systematic errors are those resulting from kinematic imperfections of the robot, for example, unequal wheel diameters or uncertainty about the exact wheelbase. Non-systematic errors are those that result from the interaction of the floor with the wheels, e.g., wheel slippage or bumps and cracks. Typically when a mobile robot system is installed with a hybrid odometry/landmark navigation system, the density in which the landmarks must be placed in the environment is determined empirically and is based on the worst-case systematic errors. Such systems are likely to fail when one or more large non-systematic errors occur. Several methods have been proposed to correct non-systematic errors occurring due to bumps, cracks or other irregularities [1] and a commercial version of a robot that incorporates such techniques is now available under the name “OmniMate”. However, even on smooth terrains, the most accurate odometric systems can at best achieve accuracies of the order of a few centimeters [1].

### **2.1.2 Active Beacon**

When the work space of the mobile robot is static, it is possible to engineer the environment in order to simplify the localization problem. Active beacons and triangulation can be used in such cases to localize the robot. As explained in chapter 1, in

order to triangulate via lateration or angulation, active beacons can be used to find distance information, e.g., LIDAR or SONAR systems, or find the geographical bearing, e.g., as in GPS systems. Although active beacon systems are considerably more accurate than odometric, they incur high costs of installation and maintenance. Accurate mounting of beacons is required for accurate positioning. Although GPS systems are extensively used for outdoor purposes (e.g, hiking, traffic flow control, etc), they are highly unreliable for indoor measurements [4].

## **2.2 A survey of locationing systems**

### **2.2.1 Active badges**

The first and arguably archetypal indoor badge sensing system, the Active Badge location system [7], which was developed at AT&T, consists of a cellular proximity system that uses diffused infrared technology. Each person the system can locate wears a small infrared badge like that shown in Fig. 2-1. The badge emits a globally unique identifier every 10 seconds or on demand. A central server collects this data from fixed infrared sensors around the building, aggregates it, and provides an application programming interface for using the data. The Active Badge system provides absolute location information. A badge's location is symbolic, representing, for example, the room—or other infrared constraining volume—in which the badge is located. As with any diffuse infrared system, Active Badges have difficulty in locations with fluorescent lighting or direct sunlight because of the spurious infrared emissions these light sources generate. Diffused infrared has an effective range of several meters, which limits cell sizes to small- or medium-sized rooms. In larger rooms, the system can use multiple infrared beacons.



**Figure 2-1:** Active badge (right) and base station (left)

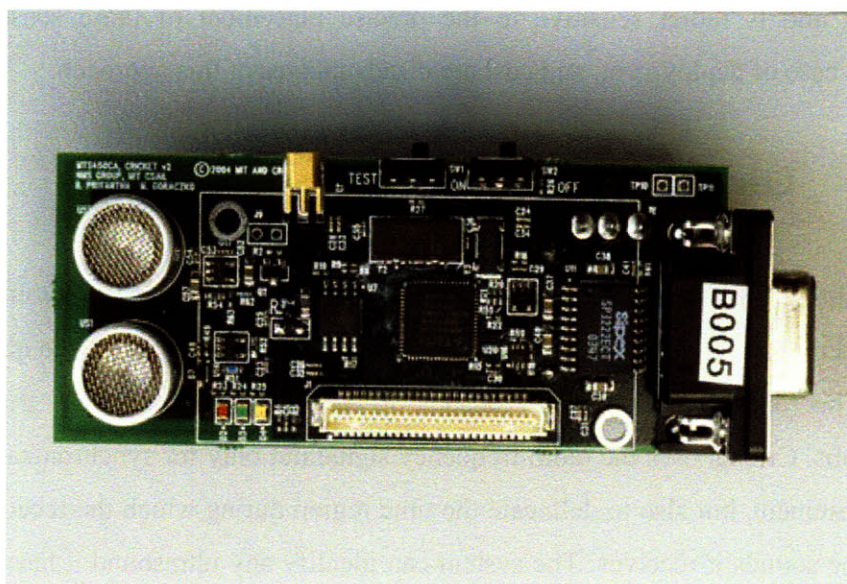
### **2.2.2 Active Bats**

The Active Bat location system [8] uses an ultrasound time-of-flight lateration technique to provide more accurate physical positioning than Active Badges. Users and objects carry Active Bat tags. In response to a request the controller sends via short-range radio, a Bat emits an ultrasonic pulse to a grid of ceiling-mounted receivers. At the same time the controller sends the radio frequency request packet, it also sends a synchronized reset signal to the ceiling sensors using a wired serial network. Each ceiling sensor measures the time interval from reset to ultrasonic pulse arrival and computes its distance from the Bat. The local controller then forwards the distance measurements to a central controller, which performs the lateration computation. Statistical pruning eliminates erroneous sensor measurements caused by a ceiling sensor hearing a reflected ultrasound pulse instead of one that traveled along the direct path from the Bat to the sensor. The system can locate Bats to within 9 cm of their true position for 95 percent of the measurements. It can also compute orientation information given predefined knowledge about the placement of Bats on the rigid form of an object and allowing for the ease with which ultrasound is obstructed. Each Bat has a GUID for addressing and recognition. Using ultrasound time of flight this way requires a large fixed-sensor infrastructure throughout

the ceiling and is rather sensitive to the precise placement of these sensors. Thus, scalability, ease of deployment, and cost are disadvantages of this approach.

### **2.2.3 Cricket**

Complementing the Active Bat system, the Cricket Location Support System [9] uses ultrasound emitters to create the infrastructure and embeds receivers into the object being located. This approach forces the objects to perform all their own triangulation computations. Cricket uses the radio frequency signal not only for synchronization of the time measurement, but also to delineate the time region during which the receiver should consider the sounds it receives. The system can identify any ultrasound it hears after the end of the radio frequency packet as a reflection and ignore it. A randomized algorithm allows multiple uncoordinated beacons to coexist in the same space. Each beacon also transmits a string of data that describes the semantics of the areas it delineates using the short-range radio. Like the Active Bat system, Cricket uses ultrasonic time-of-flight data and a radio frequency control signal, but this system does not require a grid of ceiling sensors with fixed locations because its mobile receivers perform the timing and computation functions. Cricket, in its currently implemented form, is much less precise than Active Bat in that it can accurately delineate  $4 \times 4$  square-foot regions within a room, while Active Bat is accurate to 9 cm. However, the fundamental limit of range-estimation accuracy used in Cricket should be no different than Active Bat. Cricket implements both the lateration and proximity techniques. Receiving multiple beacons lets receivers triangulate their position. Receiving only one beacon still provides useful proximity information when combined with the semantic string the beacon transmits on the radio. Cricket's advantages include privacy and decentralized scalability, while its disadvantages include a lack of centralized management or monitoring and the computational burden—and consequently power burden—that timing and processing both the ultrasound pulses and RF data place on the mobile receivers.



**Figure 2-2:** The Cricket beacon

## **2.2.4 Radar**

RADAR [10] is a building-wide tracking system based on the IEEE 802.11 WaveLAN wireless networking technology. RADAR measures, at the base station, the signal strength and signal-to-noise ratio of signals that wireless devices send, then it uses this data to compute the 2D position within a building. Microsoft has developed two RADAR implementations: one using scene analysis and the other using lateration. The RADAR approach offers two advantages: It requires only a few base stations, and it uses the same infrastructure that provides the building's general-purpose wireless networking. Likewise, RADAR suffers two disadvantages. First, the object it is tracking must support a wireless LAN, which may be impractical on small or power-constrained devices. Second, generalizing RADAR to multi-floored buildings or three dimensions presents a nontrivial problem. RADAR's scene-analysis implementation can place objects to within about 3 meters of their actual position with 50 percent probability, while the signal-strength lateration implementation has 4.3-meter accuracy at the same probability level. Although the scene-analysis version provides greater accuracy, significant changes in the environment, such as moving metal file cabinets or large groups of people congregating

in rooms or hallways, may necessitate reconstructing the predefined signal-strength database or creating an entirely new database. Several commercial companies such as WhereNet (<http://www.widata.com>) and Pinpoint (<http://www.pinpointco.com>) sell wireless asset-tracking packages, which are similar in form to RADAR. Pinpoint's 3D-iD performs indoor position tracking using proprietary base station and tag hardware to measure radio time of flight. Pinpoint's system achieves 1-3 meter accuracy and, by virtue of being a commercial product, offers easier deployment and administration than many research systems. The 3D-iD system suffers the disadvantage that each antenna has a narrow cone of influence, which can make ubiquitous deployment prohibitively expensive. Thus, 3D-iD best suits large indoor space settings such as hospitals or warehouses. It has difficulty interoperating with the 802.11 wireless networking infrastructure because of radio spectrum collision in the unregulated Industrial, Scientific, and Medical band.

### **2.2.5 Electromagnetic sensors**

Electromagnetic sensing offers a classic position tracking method [11]. The large body of research and products that support virtual reality and motion capture for computer animation often offer modern incarnations of this technology. For example, Ascension offers a variety of motion-capture solutions such as the MotionStar DC magnetic tracker (Fig. 2-3). These tracking systems generate axial DC magnetic-field pulses from a transmitting antenna in a fixed location. The system computes the position and orientation of the receiving antennas by measuring the response in three orthogonal axes to the transmitted field pulse, combined with the constant effect of the earth's magnetic field. Tracking systems such as MotionStar sense precise physical positions relative to the magnetic transmitting antenna. These systems offer the advantage of very high precision and accuracy, on the order of less than 1 mm spatial resolution, 1 ms time resolution, and 0.1° orientation capability. The main disadvantage of electromagnetic systems is that they cannot be used in a metallic environment (such as large-scale assembly) which would severely affect the magnetic field lines. Other disadvantages include steep implementation costs and the need to tether the tracked object to a control

unit. Further, the sensors must remain within 1 to 3 meters of the transmitter, and accuracy degrades with the presence of metallic objects in the environment. Many other technologies have been used in virtual environments or in support of computer animation. A CDMA radio ranging approach has been suggested [12], and many companies sell optical, infrared, and mechanical motion-capture systems. Like MotionStar, these systems are not designed to be scalable for use in large, location-aware applications. Rather, they capture position in one precisely controlled environment.



Figure. 2-3: MotionStar magnetic tracker. Transmitter antennas (left and right), receiver (center)

## 2.2.6 Computer Vision techniques

Several groups have explored using computer vision technology for localization [13, 14, 15]. Microsoft Research's Easy Living provides one example of this approach. Easy-Living [16] uses the Digiclops real-time 3D cameras (shown in Fig. 2-3) to provide stereo-vision positioning capability in a home environment. Although Easy-Living uses high-performance cameras, computer vision systems typically use substantial amounts of processing power and memory to analyze frames captured with comparatively low-complexity hardware. State-of-the-art integrated systems [17] demonstrate that



multimodal processing—silhouette, skin color, and face pattern—can significantly enhance accuracy. Vision location systems must, however, constantly struggle to maintain analysis accuracy as scene complexity increases and more occlusive motion occurs. The dependence on infrastructural processing power, along with public wariness of ubiquitous cameras, can limit the scalability or suitability of vision location systems in many applications.



**Figure. 2-3:** Digiclops 3D camera

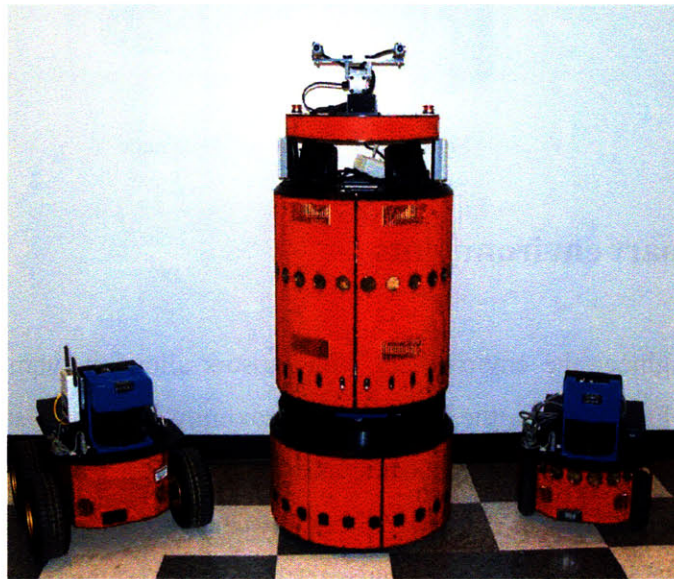
### **2.2.7 Smart environments**

Some systems engineer the environment with sensors allowing them to localize with greater accuracy. For example, the Smart Floor proximity location system [18] captures footfalls using embedded pressure sensors, and uses the data for position tracking and pedestrian recognition. This unobtrusive direct physical contact system does not require people to carry a device or wear a tag. However, the system has the disadvantages of poor scalability and high incremental cost because the floor of each building in which Smart Floor is deployed must be physically altered to install the pressure sensor grids.

### **2.2.8 Sensor Fusion**

Defined as the use of multiple technologies or location systems simultaneously to form hierarchical and overlapping levels of sensing, sensor fusion can provide aggregate

properties unavailable when using location systems individually. For example, integrating several systems with different error distributions may increase accuracy and precision beyond what is possible using an individual system. The more independent the techniques, the more effectively they can be combined. An example of current sensor fusion research, multisensory collaborative robot localization and map building presents a problem usually divided into two sub-problems: i) tracking location as the environment changes or the robot moves, and ii) determining robot location from a zero-knowledge start state. Autonomous robots, such as those shown in Fig. 2-4, employ a myriad of onboard sensors including ultrasound and laser range finders, inertial trackers, and cameras. The robots use Markov and Bayesian statistical techniques and multi-robot collaboration to accomplish sensor fusion [19]. These techniques provide important starting points for combining location systems for ubiquitous computing.

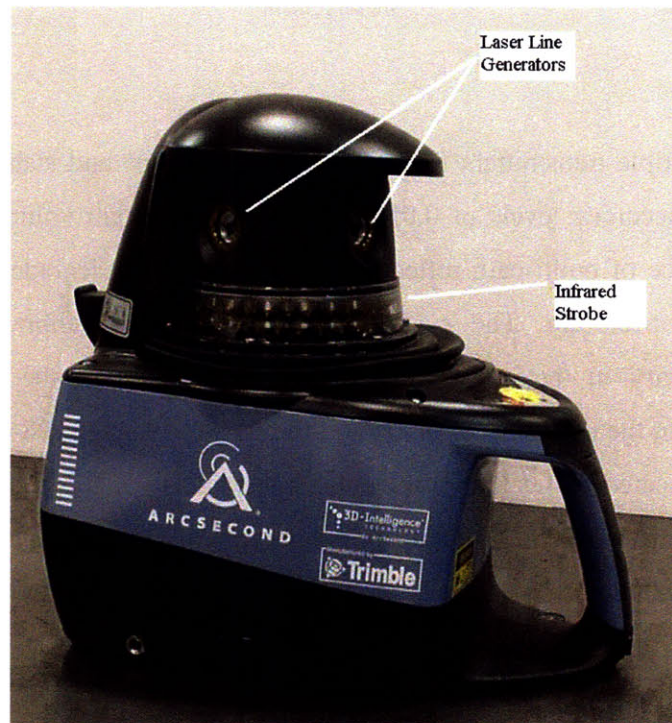


**Figure 2-4:** Robots with multiple sensors integrated for localization, multirobot collaboration and map-building.

### **2.2.9 iGPS System**

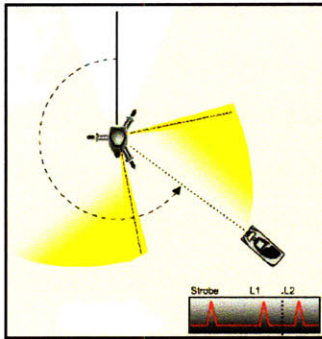
iGPS [20] uses compact infrared Laser Transmitters that are installed in facilities as infrastructure as opposed to satellites orbiting in space. Each transmitter has a rotating

head that projects two infrared line lasers at  $\pm 45$  degrees to the axis of rotation. The two lasers are spaced 90 degrees apart (about the axis of rotation) on the head. Below the rotating head is a ring of IR LEDs that are used as a timing strobe firing once per revolution (See Fig. 2-5). Sensors are mounted on the object to be tracked which detect the light signals and send it to a receiver unit for calculation of angles. The sensors wait for the timing strobe, measure the time between the two laser pulses to determine the vertical angle and the time from strobe to the laser pulses to determine the horizontal angle (see Figs. 2-6 & 2-7).

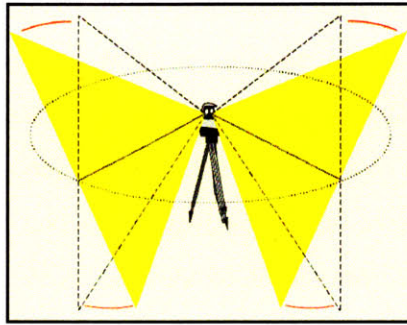


**Figure. 2-5: iGPS transmitter**

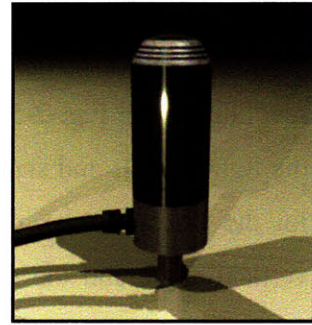
As shown in Fig. 2-8, the sensors consist of several light detectors arranged in cylindrical fashion with an IR window surrounding them. This equips them with a 270 degree field of view around the cylindrical axis and a  $\pm 30$  degree field of view in the vertical direction with a blind spot above and below each sensor. In-order to localize there need to be a minimum of 3 transmitters with a line of sight of the sensors. Once the system obtains the angular information of the sensor from at least three distinct transmitters, it can obtain the exact coordinates of the sensor using triangulation techniques.



**Figure. 2-6:** Horizontal angle measurement



**Figure. 2-7:** Vertical angle measurement



**Figure. 2-8:** iGPS receiver

Using multiple transmitters for redundant calculations and stable averaging, the system achieves accuracy levels of 0.05mm. However, in small volume environments, due to the presence of multi-path reflections; its performance degrades substantially to accuracy levels of 0.5mm. The system also has stringent calibration issues with calibration necessary in the event of even a slight change in the positions of the transmitters. Due to the geometry of lasers and the sensor design, iGPS cannot be used to localize objects located closer than 2m from a transmitter. iGPS is a highly precise but expensive localization system, the bare minimum cost of purchasing a working system being around \$180,000.

# Chapter 3

## Algorithms

### 3.1 Problem Statement

Consider the 3D environment of an indoor room as shown in Fig. 3-1. Let us suppose a mobile robot has to navigate the environment, without any collisions, to perform a high precision mechanical task (for e.g., a fitting operation, search and rescue operation etc).

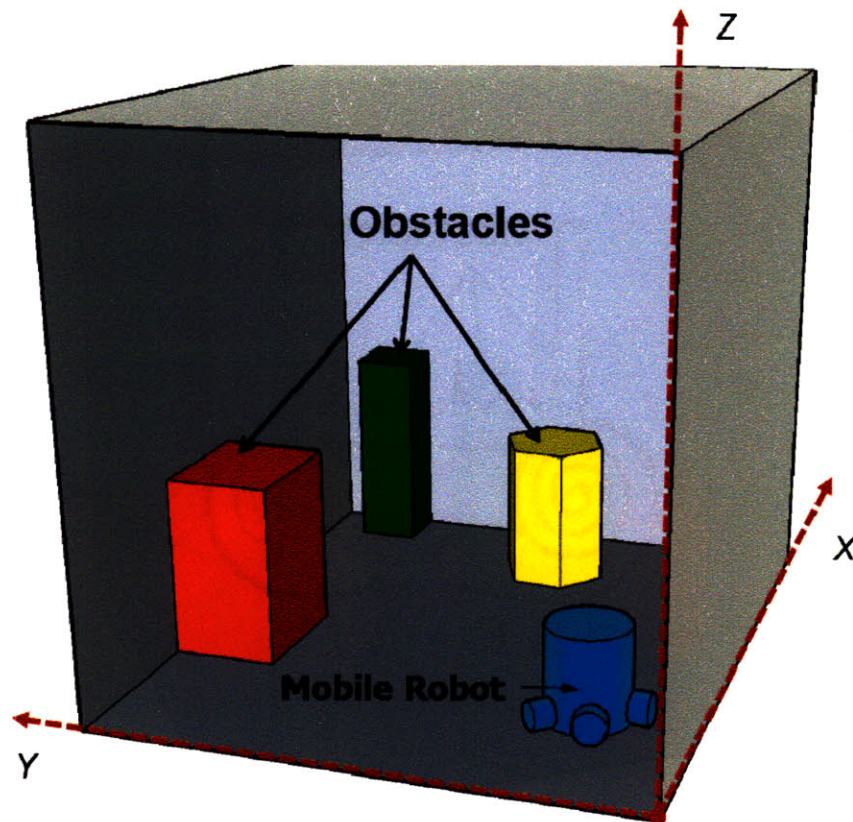


Figure. 3-1: 3D environment for localization

In order to guide the robot to the exact position, one has to know the current coordinates of the robot in a global reference. Let us suppose that the current coordinates of the robot (which are to be determined) are given by  $[x(t), y(t), z(t)]$  where  $t$  is the time index. In practice, any set of axes can be chosen with respect to which the coordinates of the robot can be conveniently calculated as long as the coordinates of the target points are precisely predetermined in such a coordinate system. Using feedback from a system that can continuously track the coordinates of the mobile robot, one can not only guide the robot to target locations, but also calibrate movements and understand the kinematics of motion.

### 3.2 Solution

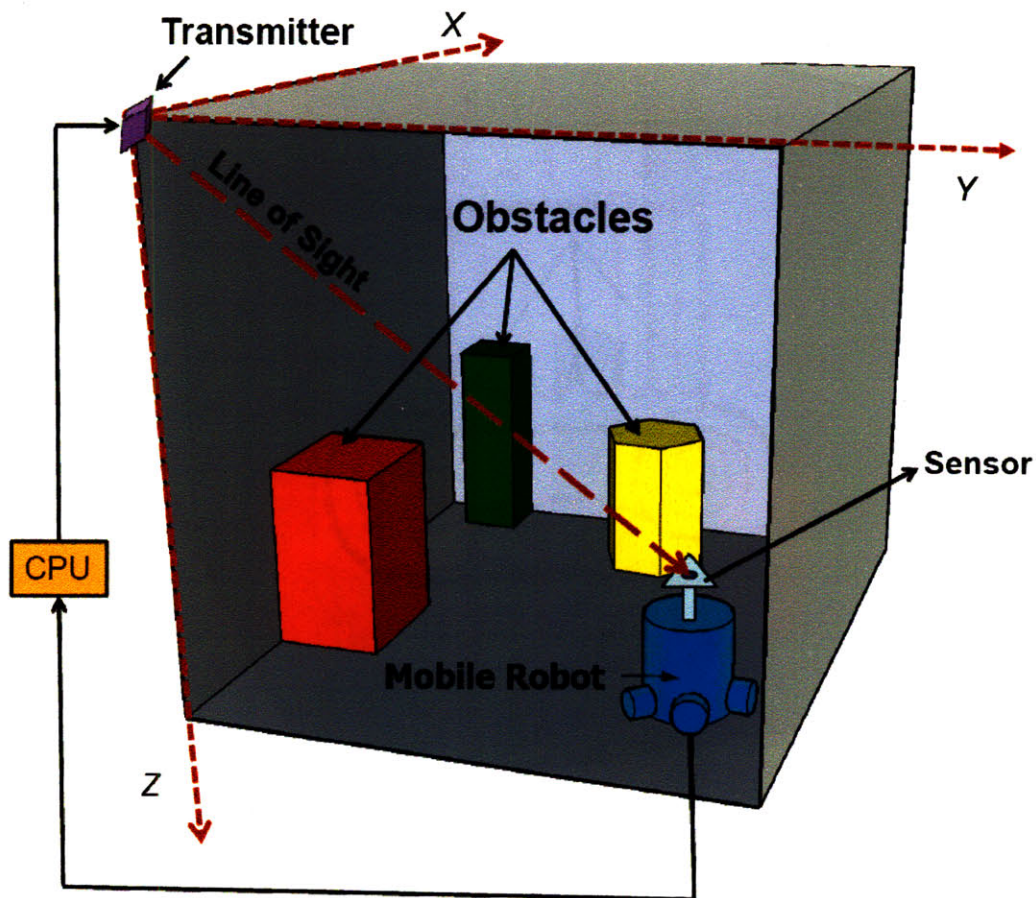


Figure. 3-2: Localization in 3D environment

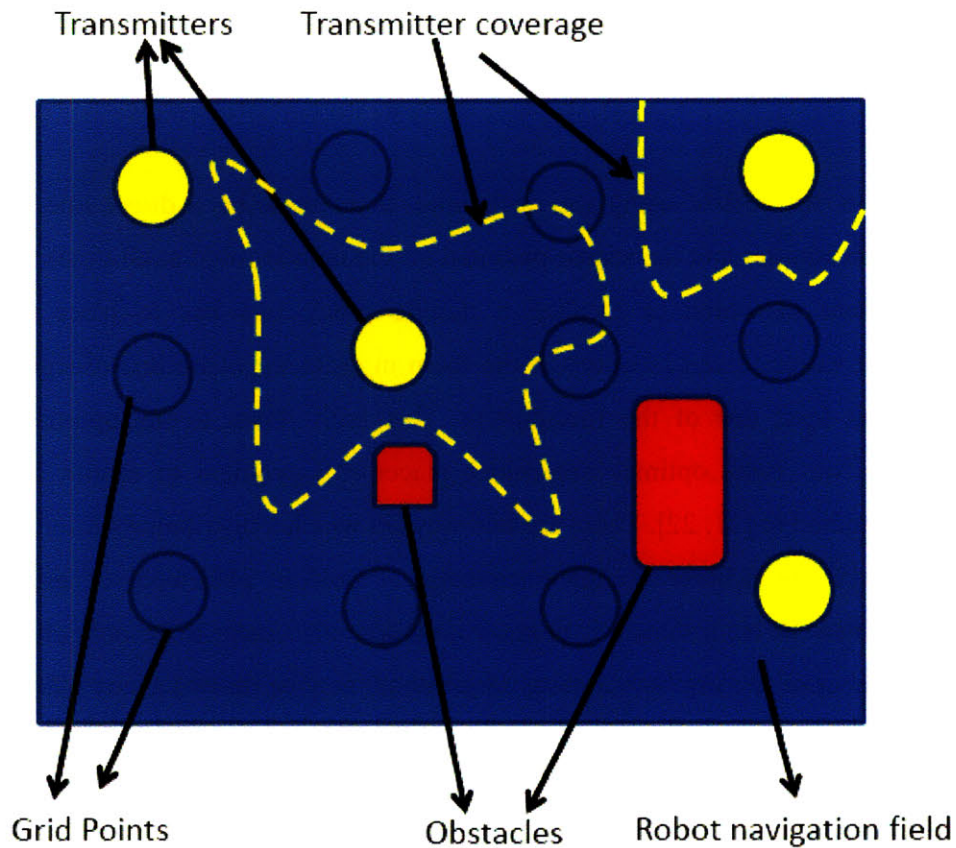
In the current thesis we propose the following solution to the localization problem described in section 3.1

- i) At a fixed position within the room, a transmitter module is placed which emits light signals detected by a sensor placed on the mobile robot.
- ii) As shown in Fig. 3-2, the origin of the global coordinate system is chosen to coincide with the center of the transmitter module. All the target locations to which the robot has to navigate in the 3D environment within the room are predetermined according to this coordinate system and remain fixed throughout the experiment.
- iii) The sensor mounted on the mobile robot detects the light signals emitted by the transmitter, and sends them to a central control and processing unit which not only calculates the coordinates of the mobile robot in the global reference frame but also controls the light signals emitted by the transmitter.

### **3.2.1 Transmitter placement in a given 3D space**

In order for us to localize using this technique, there must be a direct line of sight between the transmitter and the sensor mounted on the mobile robot at all times. Given a 3D environment in which to localize a mobile robot, one has to place multiple transmitters at various fixed locations in the room in order to ensure line-of-sight of the sensor with at least one of the transmitters. This boils down to the optimal sensor placement (in our case, optimal transmitter placement) problem to ensure complete coverage as studied in [21, 22]. There are two ways in which a distributed sensor network can be deployed to achieve complete coverage – random placement or grid based placement. When the environment is unknown, random placement is the only choice and sufficient number of sensors (in our case, transmitters) may be thrown at any place within the environment ensuring coverage. However, in our problem, since the properties of the environment are predetermined, optimal placement points can be calculated. In order to solve this problem, we make the following assumptions (see Fig. 3-3):

- i) The field is generally divided into grid points and it is to be determined whether a transmitter is needed to be placed at a particular grid point. In practice, these grid points represent the various points within the room where it is feasible for the transmitters to be placed.
- ii) The transmitter coverage pattern has to be carefully measured when it is placed at each of the grid points. The coverage of a transmitter is defined as the set containing all points of the room from which the sensor mounted on the mobile robot would successfully detect the light signals emitted by the transmitter. Hence, depending on the geometry of the room, obstacles in the navigation path and features in the environment, the coverage pattern of the transmitter would be different at each of the grid points.
- iii) If every point in the robot navigation field can be detected by at least one transmitter, we call the field completely covered.



**Figure. 3-3:** Sensor field with grid points



Then, one possible solution to the coverage problem can be given by the following algorithm:

- 1) Form all possible subsets of grid points such that each subset has the property that with transmitters placed at grid points contained in it, the subset can cover the entire navigation field.
- 2) From the above subsets, choose the subset with the *minimal* cardinality and place the transmitters at the grid points contained in the chosen set.

Such an algorithm would ensure the placement of minimal number of transmitters in the room guaranteeing complete coverage.

### **3.3 Transmitter Identification**

Once the optimal grid points for transmitter placement are chosen using the algorithm presented in section 3.2.1, the central processing unit has to identify the transmitter from which the robot is currently receiving the light signals. This is important because calculations based on the light signals received from different transmitters would give us the coordinates in different global axes of reference each of which is centered at its respective transmitter. Two possible techniques could be adapted in such a scenario to identify the transmitter:

- 1) At the end of each measurement, scanning pulses are transmitted by each transmitter within the selected subset one at a time in a predefined sequence. The CCPU then selects the transmitter whose scanning pulse was correctly received by the robot to be the active transmitter for the next set of measurements. In the case of more than one transmitter pulses being detected by the robot, one can be randomly chosen.
- 2) Alternatively, if the transmitters are composed of any rotating devices (for e.g., rotating laser beams), one could chose different speeds of rotation for the various transmitters and based on the received light pulses, determine the transmitter covering the robot.

### 3.4 Solution to the localization problem

The system we propose in this thesis for indoor localization consists of the following components:

- i) Transmitter module which consists of rotating line lasers.
- ii) Sensor (or equivalently, the receiver) which is mounted on the mobile robot. The sensor consists of multiple photodiodes which receive the light signals emitted by the transmitter.
- iii) Central Control & Processing Unit (CCPU) consists of digital circuitry built on Xilinx FPGA
- iv) Feedback control unit which controls the movements of the robot based on the readings of the localization system.

A higher level block diagram of the localization system is presented in Fig. 3-4.

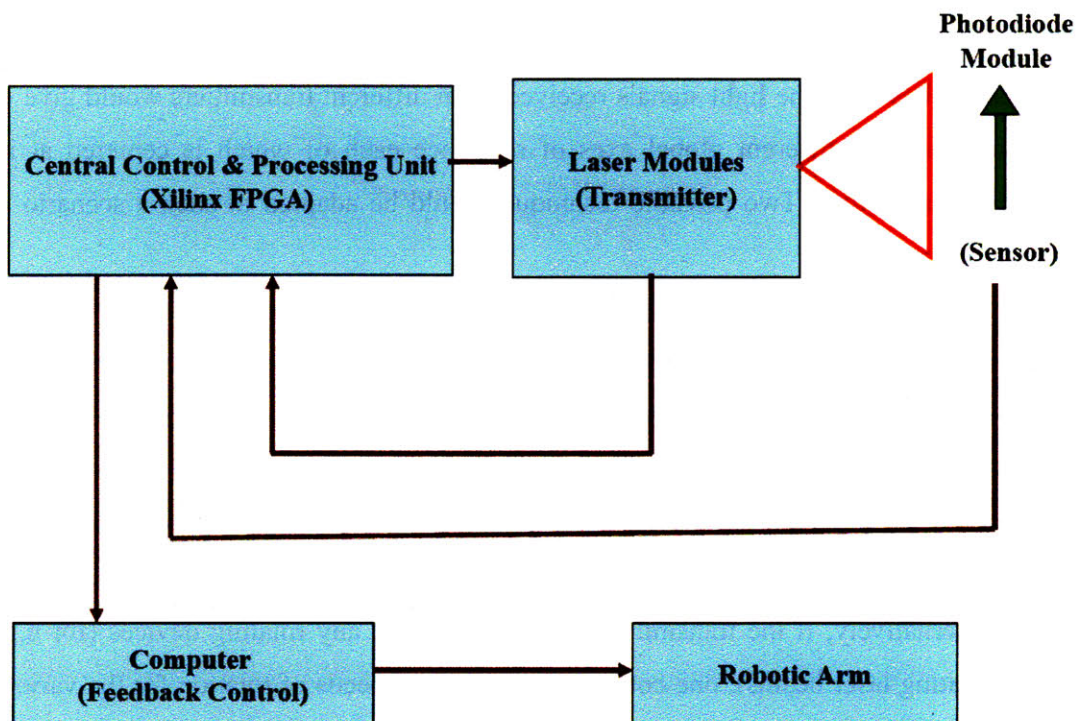


Figure. 3-4: Block diagram of the localization system

## 3.5 Localization technique

The transmitter consists of three rotating line lasers mounted in a straight line as shown in Fig. 3-5. The rotating line lasers are used to accurately measure angles which in turn, are used to calculate the coordinates of the sensor (receiver) mounted on the mobile robot. The reason for choosing the straight line configuration is as follows: knowing i) the angles of the receiver from laser A and laser C, i.e.  $\theta_1$  and  $\theta_3$  (see Fig. 3-6), and ii) the straight line distance between the lasers, fixes the values of two orthogonal coordinates ( $x$  and  $y$  in Fig. 3-6) of the receiver and restricts the locus of possible points to a straight line perpendicular to the  $x$ - $y$  plane thus leaving the  $z$ -coordinate as the only degree of freedom. This follows from a simple 2D geometrical fact: the base length and the corresponding base angles are enough to locate the vertex of a triangle. In order to find the target point on the straight line locus, one only needs to know another angle measured in a plane orthogonal to the plane containing the first two angles  $\theta_1$  and  $\theta_3$  (i.e., the  $x$ - $y$  plane). This angle is measured by laser D (angle  $\theta_2$ ). Thus, knowing the values of  $l$ ,  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  one can uniquely determine the 3D coordinates of the receiver. Note that for this technique to work, the center of the global coordinate system should coincide with the midpoint of the straight line containing the three rotating line laser modules, i.e., with the center of laser D.

In order to determine the coordinates of the receiver using the above procedure, one needs to find all the three angles of the receiver fixed at a particular position. Hence, line of sight of the receiver with all the three lasers is quintessential. Also, due to power constraints, the lasers have limited range. Therefore, even if there exists a line of sight, the intensity of the lasers may not be high enough for detection by the photodiode. Hence, given an unknown 3D environment with obstacles and grid points (see section 3.2.1) where the transmitters (a transmitter consists of all the three lasers placed in a straight line) could be placed, one can place a transmitter at each of the grid points, physically measure the area of coverage of each transmitter (area of coverage is determined by both line of sight and intensity constraints) and then choose the coverage

subset with minimal cardinality of grid points (see section 3.2.1). Rules similar to those described in section 3.3 could then be used for transmitter identification.

The sensor (receiver) module consists of four photodiodes arranged in specific pattern to detect the laser beam and transmit the signals to the central control and processing unit. The hardware used to build the laser and sensor modules is described comprehensively in chapter 4 of the thesis.

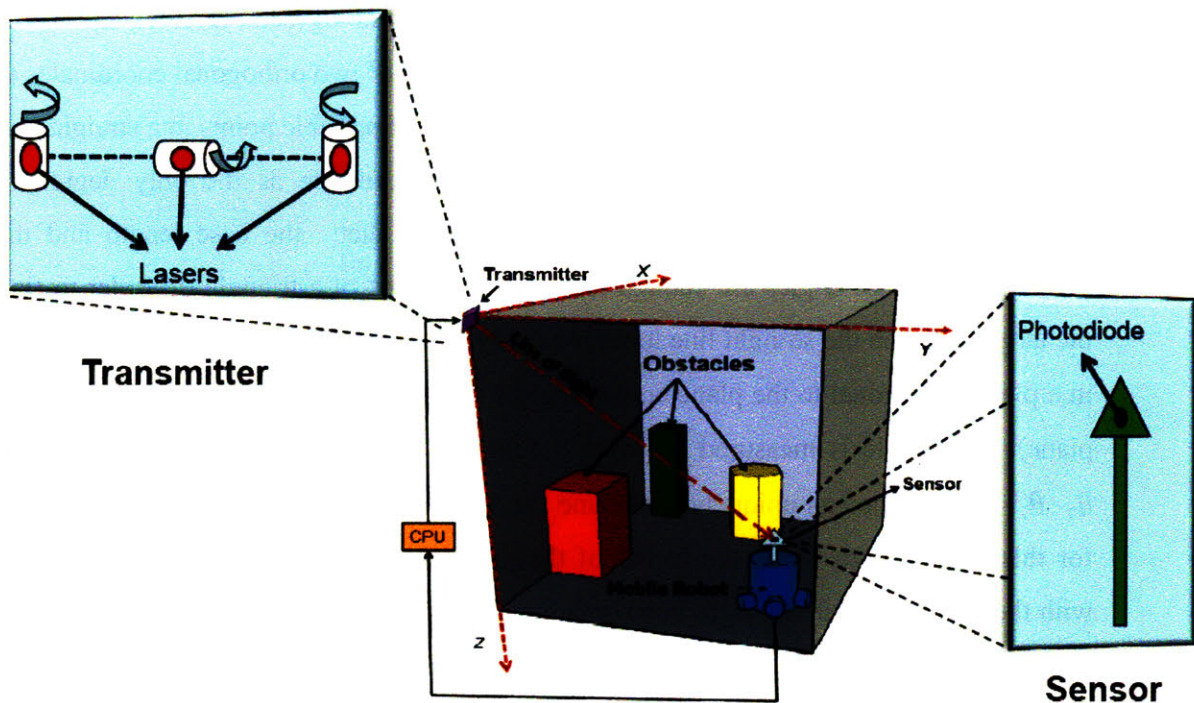
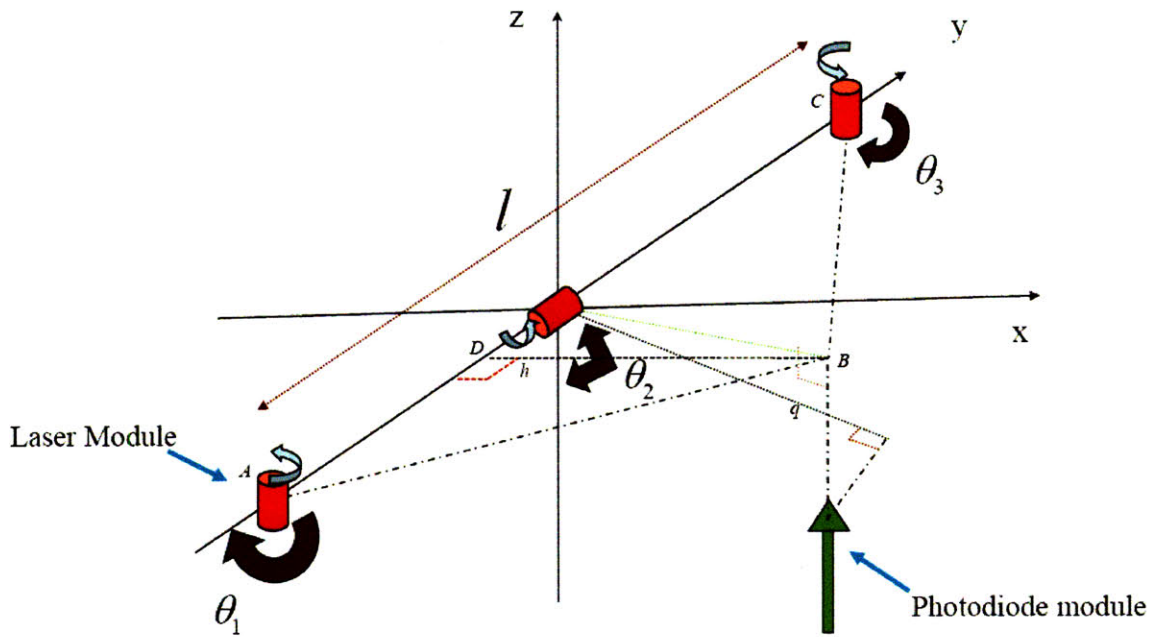


Figure. 3-5: Transmitter and receiver sensor modules

The line lasers are mounted on DC motors rotating continuously. The state of the line lasers (on/off) is controlled by the central control and processing. A particular fixed point on the shaft of the motor is taken as the zero reference point and the angular rotation of the motor with respect to this point is recorded at the moment the laser strikes the photodiodes on the sensor. Thus, as shown in Fig. 3-6, one can obtain three distinct angles of the sensor, one from each of the rotating line lasers. The lasers are mounted so that their line emissions are parallel to the vertical axis of the cylinder. Hence, in Fig. 3-6, the line laser beams emitted by laser modules A and C are parallel to the z-axis and the beam emitted by module D is parallel to the y-axis.



**Figure. 3-6:** Angle measurements for localization

The coordinates  $(x, y, z)$  of the sensor in the coordinate system shown in Fig. 3-6 can be determined by simple trigonometric rules as follows:

$$\overline{AD} = h \cot(\pi - \theta_1) = -h \cot(\theta_1)$$

$$\overline{CD} = h \cot(\pi - \theta_3) = -h \cot(\theta_3)$$

Since,

$$l = \overline{AD} + \overline{CD}$$

we have,

$$l = -h(\cot \theta_1 + \cot \theta_3)$$

$$h = \frac{-l}{(\cot \theta_1 + \cot \theta_3)} \quad (3.1)$$

$$x = h \quad (3.2)$$

$$y = \frac{l}{2} + h \cot \theta_3 \quad (3.3)$$

For the z-coordinate, we use the following relation,

$$q \sin \theta_2 = h$$

Therefore,

$$q = \frac{h}{\sin \theta_2}$$

And,

$$z = -q \cos \theta_2$$

Hence,

$$z = -h \cot \theta_2 \quad (3.4)$$

If the sensor is present in the region  $\theta_2 > 180^\circ$ , the equations are given by:

$$h = \frac{-l}{(\cot \theta_1 + \cot \theta_3)}$$

$$x = -h \quad (3.5)$$

$$y = \frac{l}{2} + h \cot \theta_3 \quad (3.6)$$

and,

$$z = -h \cot \theta_2 \quad (3.7)$$

The central control and processing unit controls the on/off state of each of the lasers such that only one of the rotating lasers is active (on) at a given time. Hence the three angles are measured in a sequential order according to the following algorithm:

**ALGORITHM 1** *Localization Algorithm*

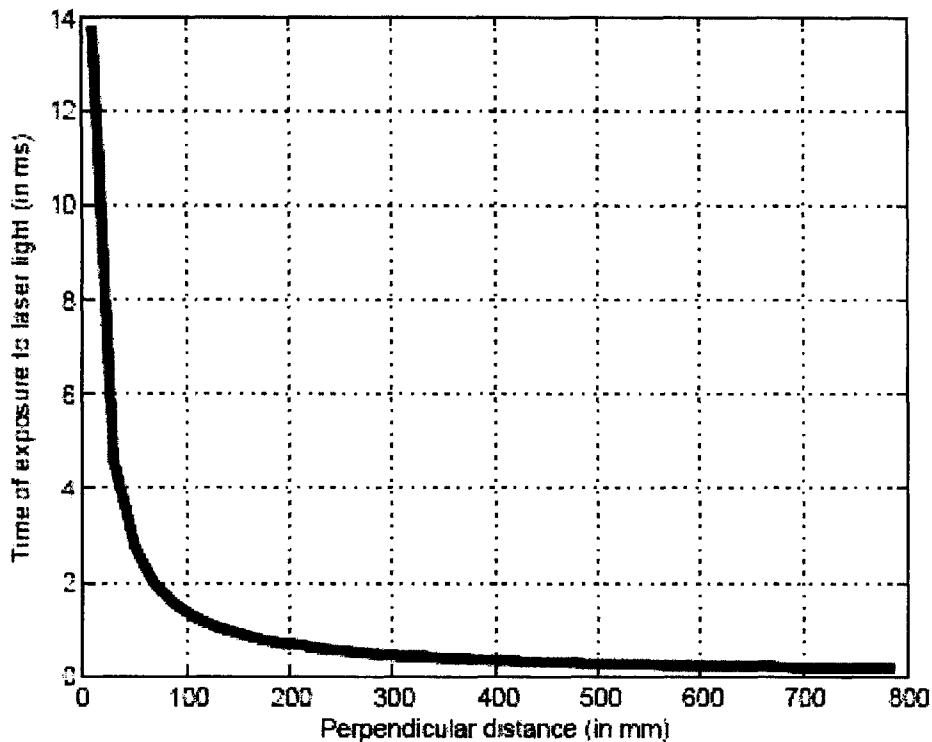
```
1:  procedure LOCALIZE(void)
2:    while(1) do
3:      for i = 1 to 3 do
4:        switch on laser(i)
5:          measure angle ( $\theta_i$ )
6:        switch off laser(i)
7:        if (i=3) do
8:          calculate (x, y, z) using ( $\theta_1, \theta_2, \theta_3$ )
9:        end if
10:     end for
11:   end while
12: end procedure
```

Once the laser is turned on after step 4 in the above algorithm, using the signal from an optical sensor mounted on the motor as reference, a state machine built in the central control and processing unit waits for one full revolution of the motor (see chapter 5) before turning off of the motor in step 6. Because one full revolution of the motor ( $360^\circ$ ) with its laser turned on guarantees the striking of the laser beam on the photodiode receiver, the required angle ( $\theta_i$ ) can be calculated as described in chapter 5.

An alternate technique of determining the coordinates of the photodiode sensor could be designed using only two laser modules to measure two angles and using the exposure time of the photodiode to laser beam to calculate the perpendicular distance from the origin to the photodiode receiver module. The exposure time of a circular photodiode to a laser beam decreases inversely proportional to distance between them (see Fig. 3-7). Though such a system seems slightly compact than the system we previously described in this section, in our practical experiments with the system, we have realized that:

- i) The system is highly sensitive to ambient AC light impinging on the photodiodes because even a slight error in measurement of distance, significantly effects the  $(x, y, z)$  coordinate values thus deteriorating the system accuracy. And,
- ii) For distances above 500 mm, the change in exposure time with increase in distance becomes very minute (less than 0.2 ms) and difficult to differentiate, therefore making the distance measurements indiscernible and at the same time more sensitive to oscillations due to ambient AC light.

One can think of improving the system performance by reducing the effect of AC oscillations (may be use the system in a dark environment or only in the presence of DC light) and by using very high frequency counters to discern the changes in exposure times for distances greater than 500 mm.



**Figure. 3-7:** Exposure time of the laser beam shining on photodiode surface Vs distance between them



# Chapter 4

## Hardware platform

In this chapter we describe the various hardware components used in building the localization system proposed in chapter 3.

### 4.1 Hardware components

The localization system consists of 3 main components (see Fig. 3-4): i) the transmitter (laser module), ii) sensor module and iii) the central control and processing unit. In this chapter we present detailed descriptions of the hardware realizations of each of the above listed components.

#### 4.1.1 The laser module

The laser module consists of a line laser emitter (Fig. 4-1) built with an integrated quartz cylindrical lens, collimating lens, laser diode and an APC driver circuit with an operating wavelength of 635 nm. The emitting angle of the laser, as shown in Fig. 4-1, is greater than 90 degrees. The laser emissions are not eye safe and hence require the continuous usage of protective glasses throughout the experiment.

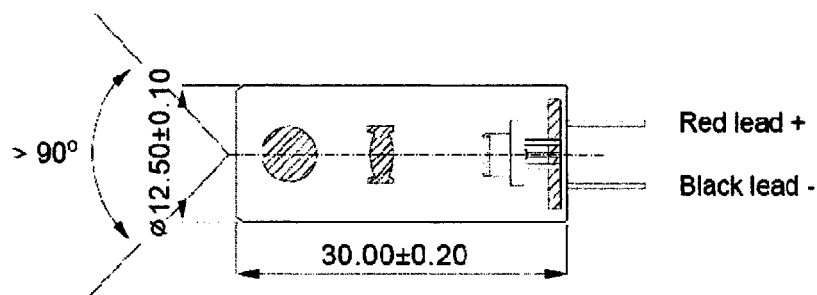
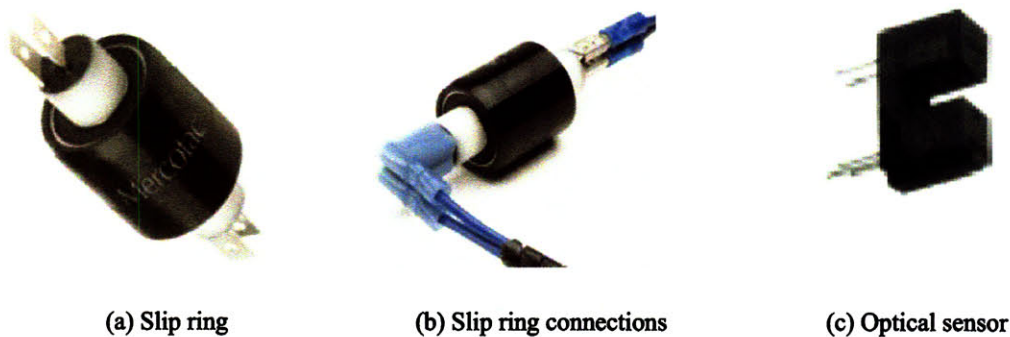


Figure. 4-1: Line laser emitter used in building the laser transmitter modules

The line laser emitter is mounted on a DC motor equipped with quadrature optical encoders. The quadrature optical encoders output 5126 square pulses each for one complete revolution of the motor. The outputs of the encoders can be passed through a XOR gate (modulo-2 addition gate) in order to double the resolution to 10252 square pulses. In addition, counting on both the rising and falling edges of the encoders output would increase the resolution to 20504.

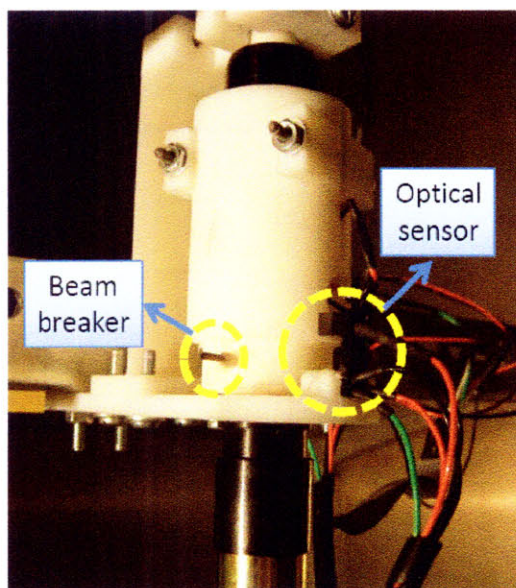
In order to control the state of the laser beams (on/off), one has to switch its power supply. To achieve a continuous rotary electrical contact with the laser diodes, we have used electrical slip rings shown in Figs. 4-2 (a) & (b). These slip-rings offer an extremely low resistance electrical connection because the electrical conduction path is a liquid metal which is molecularly bonded to the contacts. Unlike brush slip rings which are composed of a rotating ring metal upon which another graphite or metal brush rubs and transfers the electrical signal or current, the liquid metal slip-rings are robust, have low wear and tear, resistant to oxidation, have a constant electrical resistance and cause extremely low electrical noise.



**Figure. 4-2.** Slip rings and optical sensor used to build the laser transmitter module

A reference point in the revolution of the lasers is used to calculate the angular distance travelled by the laser beam at the time of striking the surface of the photodiode. An optical sensor mounted on the motor shaft serves the purpose of providing this reference point. The optical sensor (shown in Fig. 4-2 (c)) is composed of an in-built LED, preamplifier chip and photo receiver circuit which outputs a logic level square pulse whenever an opaque object blocks the line-of-sight between the LED and the photo receiver circuit. A thin cylindrical rod mounted (see Fig. 4-3) on the rotating line laser

part cuts through the opening slot of the optical sensor blocking the LED's line-of-sight and thus generating a square pulse at the end of one full revolution of the laser.



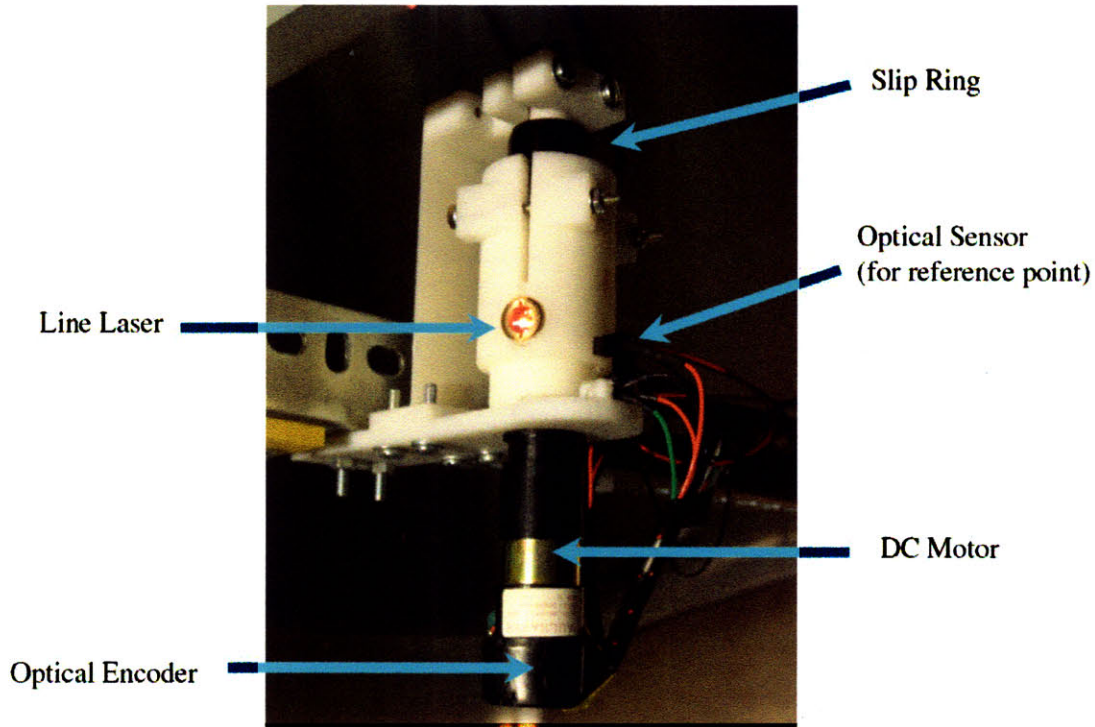
**Figure. 4-3:** A *beam breaker* would cut through the slit of the optical sensor which then produces a square pulse indicating the starting point of the revolution

A complete module with each of its individual components labeled is shown in Fig. 4-4. An amplifier circuit used to transmit the signals from the encoders to the central control and processing unit is presented in Appendix A.

### **4.1.2 Photodiode sensor module**

The photodiode sensor module (receiver) consists of four square shaped silicon photodiodes shown in Fig. 4-5 (a). The photodiodes are mounted in grooves made on the four faces (excluding the base) of a tetrahedral structure which ensures a 360 degree field of view for the sensor. The receiver is mounted on the mobile robot using two drilled holes at the bottom as shown in Fig. 4-5 (b). The design ensures that at any given point of a time, at least one of the photodiodes is visible to the rotating line lasers irrespective of the orientation of the mobile robot in the room. If more than one photodiodes is

activated by the laser beam, an equivalent of the center of mass algorithm (described in chapter 5) is used to calculate the coordinates of the centroid of the receiver.

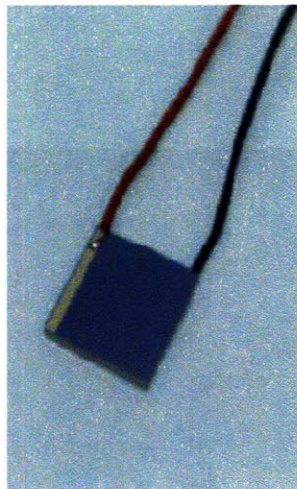


**Figure. 4-4:** Laser module with all its component parts. The transmitter consists of three such laser modules placed in a straight line

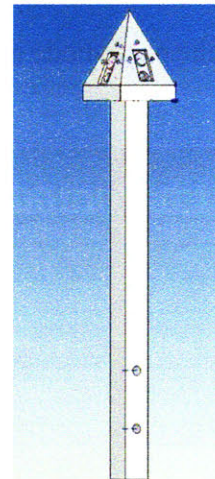
The circuitry used to receive the signals from the four photodiodes, filter, amplify and transmit them to the central control and processing unit is presented in Appendix A. Photodiode response is significantly affected by ambient light. A bright DC light source (a bright LED for example), would cause the high precision amplifier to saturate. This effect can be circumvented by carefully biasing the amplifier and adjusting its gain such that the output is not saturated when exposed to ambient light.

When the laser beam strikes the surface of the photodiode, a constant threshold comparator is used to convert its response into a square pulse with sharp edges which are later used to calculate the duration for which the laser was shining on the photodiode. Due to the juxtaposition of the response pulse and a sinusoidal wave, the presence of an AC ambient light source (a fluorescent tube for example), would cause the response of

the photodiode to oscillate. This effect would introduce errors in the timing measurements and the amount of error is proportional to the intensity of the ambient light. In order to circumvent this effect, a high pass circuit (see Appendix A) with appropriate DC bias is used to filter frequencies below 140 Hz (the frequency of ambient light being approximately 120 Hz) and hence reducing the effect of ambient light on timing measurements.

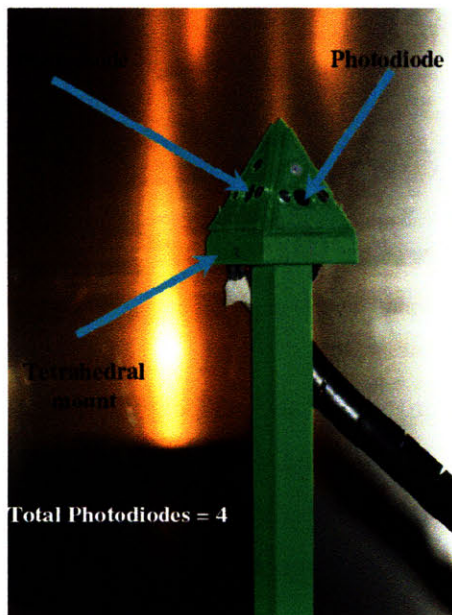


(a) Photodiode sensor



(b) Photodiode mount

**Figure. 4-5**



**Figure. 4-6:** Receiver module with photodiodes on 4 sides ensuring a 360° field of view

### 4.1.3 Central control and processing unit

The central control and processing unit consists of digital circuitry build on a programmable Xilinx FPGA board (shown in Fig. 4-7). The circuitry consists of 3 three-to-one multiplexers, four 32 bit counters and a state machine which generates the control signals for the multiplexer and the lasers. Each of the 4 counters is dedicated to one of the four LEDs in the photodiode receiver module.

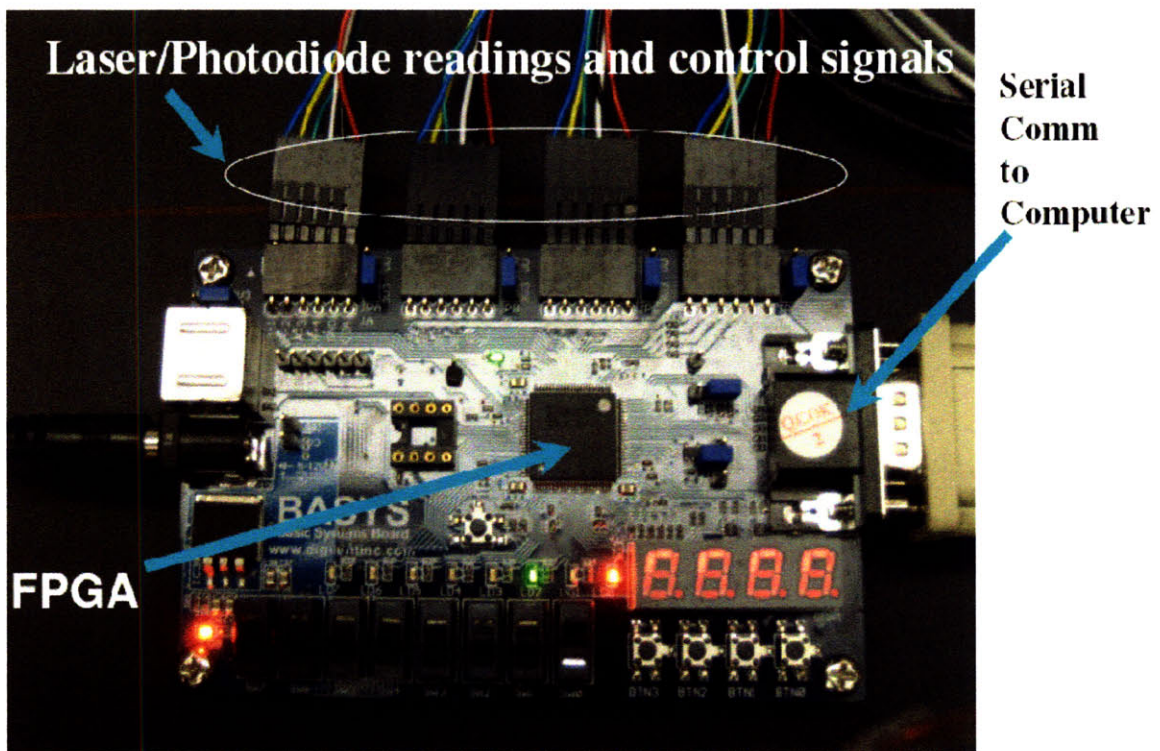
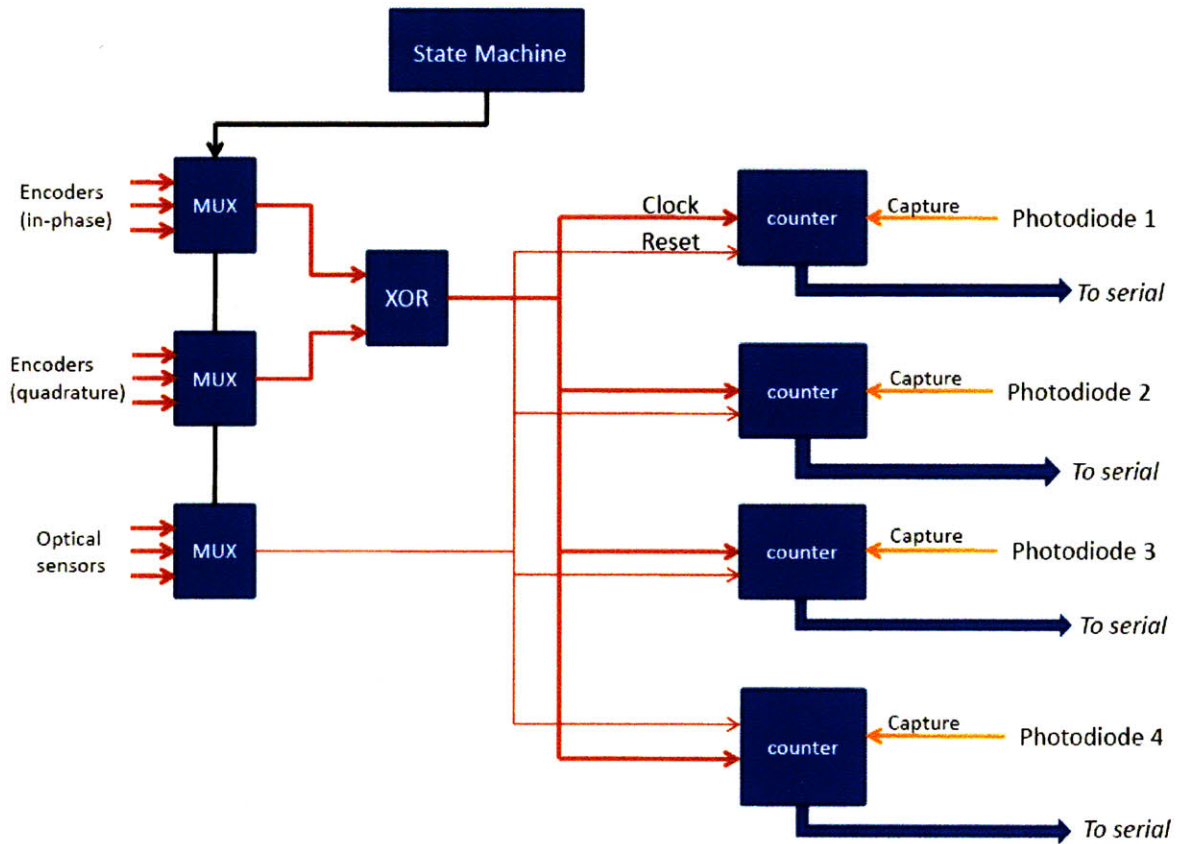


Figure. 4-7: The central control & processing unit (Basys FPGA development board)

When the laser beam strikes the surface of a photodiode, the rising and falling edges of the square pulse output are used as trigger edges to capture and store the readings of a counter whose clock consists of pulses from the encoders attached to the motors. The inputs to multiplexer 1 consist of the in-phase encoder outputs from the three laser modules, while the quadrature outputs of the encoders and the outputs from the 3 optical sensors form the inputs to multiplexer 2 and 3 respectively. The state machine

(described in section 5.3) outputs control signals such that when a particular laser module is turned on, the multiplexers output only the encoder and optical sensor pulses corresponding to that laser module. Following the measurement of angular orientation of the photodiode sensor (say  $\theta_1$ ), the state machine issues signals to turn the laser off, switch the multiplexer outputs and turn on the power to the laser in the new laser module in order to measure the next angle (say  $\theta_2$ ). All three angles ( $\theta_1, \theta_2, \theta_3$ ) are measured in this way in order to update the position of the photodiode sensor module at any given moment.



**Figure. 4-8:** Circuit block diagram of central control and processing unit

A XOR gate following the outputs from multiplexers 1 and 2 is used to modulo-2 add the in-phase and quadrature outputs from the encoders in order to increase the number of pulses per one full revolution of the lasers (and hence effectively, the resolution) from 5126 to 10252. Angular resolution is directly proportional to the number of pulses per full revolution and hence a higher number pulses is desirable. For example, with 5126 pulses one achieves an angular resolution of  $0.07^\circ$ , whereas with 10252 pulses, one achieves an angular resolution of  $0.035^\circ$  (increased by a factor of 2). The 32 bit counters are reset at the rising edge of the pulse from the optical sensor mounted on the currently active laser module. The output from the XOR gate serves as the clock signal to the counters which capture and store their respective readings on the rising and falling edges of the photodiode pulse. A higher level block diagram of the central control and processing unit is presented in Fig. 4-8 and the Verilog code for the implementation of this circuit is presented in the Appendix B.

The readings of the four counters are transmitted to a computer through a serial port on board the BASYS FPGA development board. A java program is then used by the machine to read the serial port, calculate the coordinates and plot them in real-time. The java code and the MATLAB implementation of the GUI are presented in Appendix B.



# Chapter 5

## System Design

In this chapter, we give an overview of the various software layers in the localization system, the communication flow between them and describe the computational techniques used to calculate the angles and the coordinates of the photodiode sensor.

### 5.1 System overview

The localization system consists of a central control and processing unit, transmitter modules (three rotating lasers), a receiver module and a computer (see Fig. 3-4). The various software programs running on each of these systems and the communication flow between them is shown in Fig. 5-1. A Verilog code (see Appendix B) is used to program the FPGA and build the circuit presented in Fig. 4-8. The FPGA communicates the readings of the counters to a computer using a serial communication port. A Java code (see Appendix B) running on the computer is used to read the serial port and communicate it to a MATLAB program (see Appendix B) which calculates and plots the coordinate values. These coordinate values are then communicated to the control system of the robotic arm over Ethernet which in turn controls the hexapod and brakes of the robotic arm (see chapter 6) to adjust its position.

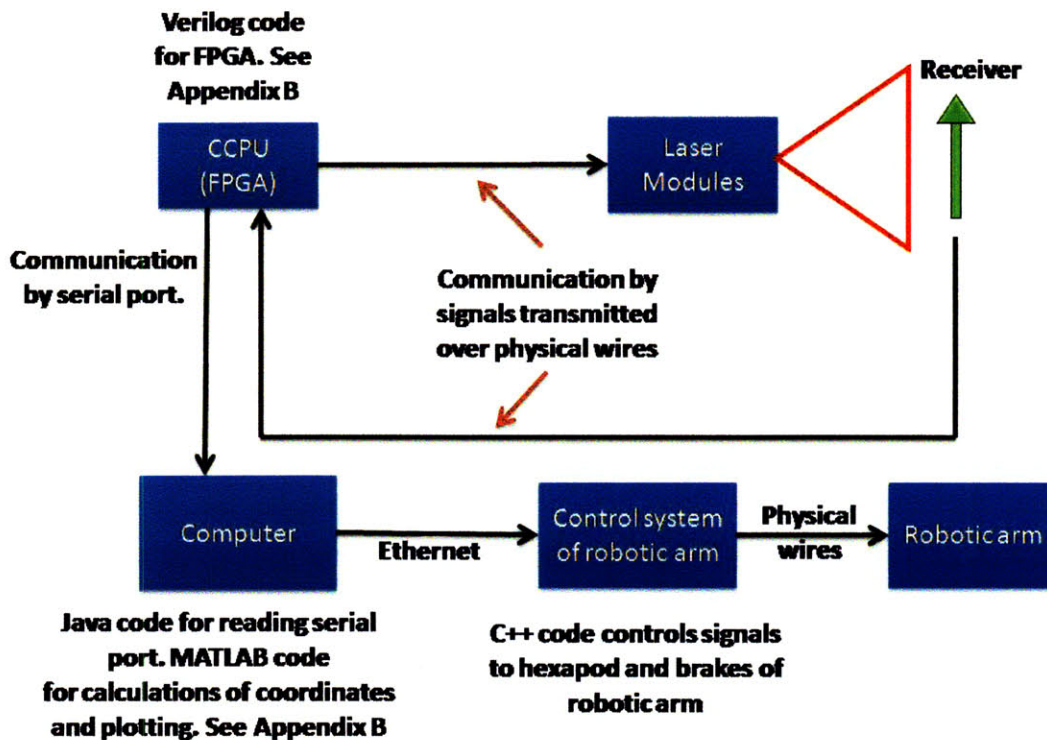


Figure. 5-1: System block diagram of localization system showing various software layers and the communication flow between them

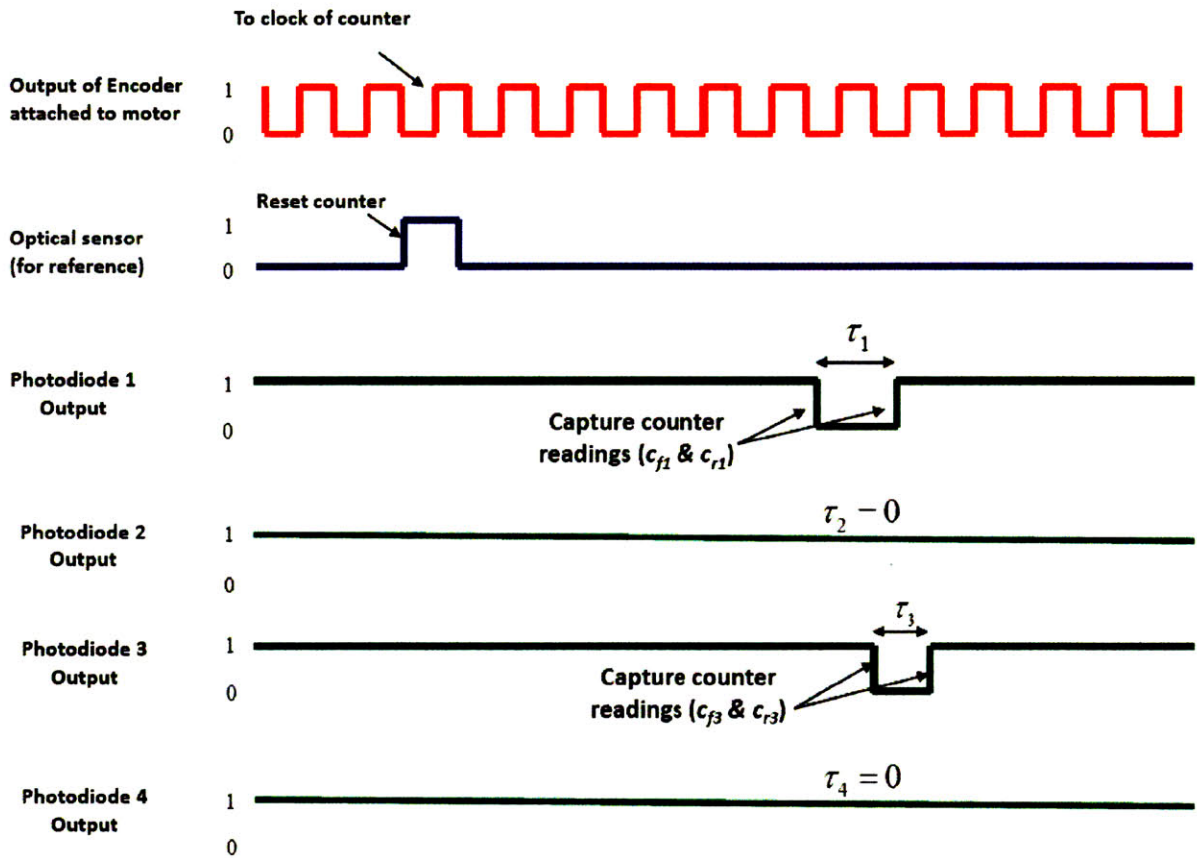
## 5.2 Angle computation

A state machine (see Figs. 4-8 and 5-3) controls the select signals to the multiplexers such that the clock to the counters is the output of the encoder attached to the active laser module modulo-2 added with its quadrature counterpart. The output of the optical sensor attached to the laser module selected above is used to reset the readings of the counter (see Fig. 5-2). The various states of the state machine are explained in Fig. 5-3.

As described in chapter 4, each photodiode in the photodiode receiver module is connected to a 32-bit counter. As the line laser beam rotates sweeping across the space, depending upon the orientation of the mobile robot, multiple photodiodes may get activated. For example, in Fig. 5-2, photodiodes 1 and 3 are exposed to the laser beam while photodiodes 2 and 4 are shadowed. The readings of the counters are captured on both the falling and rising edge of the photodiode pulse. Let  $c_{fj}$  and  $c_{rj}$  denote the

captured values of counter  $j$  on the falling and rising edges of photodiode  $j$  respectively. Therefore, in Fig. 5-2,

$$c_{f2} = c_{r2} = c_{f4} = c_{r4} = 0 \quad \& \quad (c_{f1}, c_{r1}, c_{f3}, c_{r3}) > 0$$



**Figure. 5-2:** Photodiode receiver signals. Indicated also are the various hardware/software functions performed using them.

If  $N$  denotes the number of pulses produced by the encoder (after modulo-2 addition of in-phase and quadrature components), then the angle of a particular photodiode  $j$  (for  $j = 1$  to 4) is given by:

$$\alpha_j = \left( \frac{\frac{(c_{fj} + c_{rj})}{2}}{N} \right) \times 360^\circ \quad (5.1)$$

Let the time interval (duration) for which the laser beam shines on particular photodiode  $j$  be denoted by  $\tau_j$ . Thus, for the example scenario shown in Fig. 5-2,

$$\tau_2 = \tau_4 = 0 \ \& \ (\tau_1, \tau_3) > 0$$

Also, note that,

$$\tau_j \propto (c_{rj} - c_{fj})$$

In fact, for some constant  $K$ ,

$$\tau_j = K \cdot (c_{rj} - c_{fj}) \quad (5.2)$$

Having computed the angles of the four photodiodes according to equation 5.1, the angle of centroid of the tetrahedral receiver is computed according to an equivalent of center of mass computation as follows:

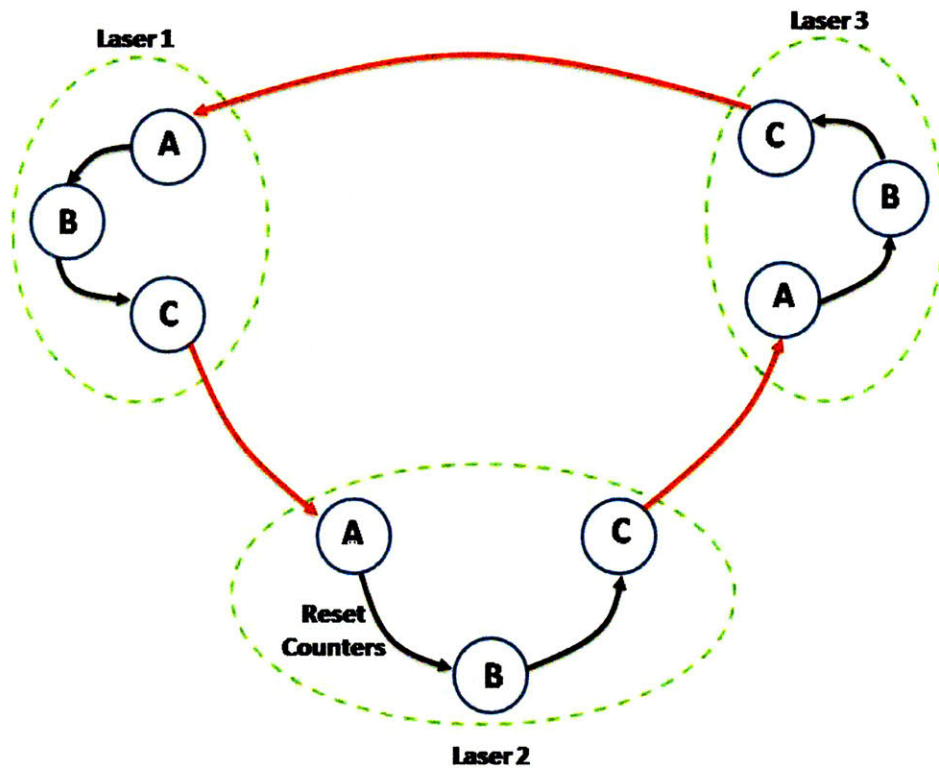
$$\theta = \frac{\sum_{j=1}^{j=4} \tau_j \cdot \alpha_j}{\sum_{j=1}^{j=4} \tau_j} \quad (5.3)$$

The computation in equation 5.3 is performed for each of the three laser modules in order to calculate the angles  $(\theta_1, \theta_2, \theta_3)$  which are required to perform triangulation as described in chapter 3. Also, depending on the speed of rotation of the motors, there exists an upper limit on the speed at which the mobile robot can move inside a room owing to the necessary requirement of calculating all the three angles for positioning.

## 5.3 State machine design

The state machine which controls the select signals to the multiplexers and the power to the lasers has three major states: *laser 1*, *laser 2* and *laser 3*, one corresponding to each laser (see Fig. 5-3). When in a particular state *laser i*, the power to laser *i* is turned on. Each of the major states further consists of three sub-states, namely states *A*, *B* and *C*, whose function is as follows:

- i) State *A*: Wait for the positive-going edge of the optical sensor. When up-going edge occurs, reset the counters and transition to state *B*.
- ii) State *B*: Wait for the positive-going edge of the optical sensor. When the positive-going edge occurs, transmit the values captured on four counters (values are captured on both the positive and negative going edges of the four photodiodes) via the serial port to the computer and transition to state *C*.
- iii) State *C*: Wait for the serial communication to complete. Upon completion, turn off the current laser, turn on the next laser, switch the outputs of the multiplexers and transition to the next major state (i.e., one of the states *laser 1*, *laser 2* or *laser 3*).



**Figure. 5-3:** The state machine built inside the central control and processing unit. It has three major states (*laser 1*, *laser 2* and *laser 3*) each consisting of sub-states *A*, *B* and *C* (see section 5.3 for description).

# Chapter 6

## Experiments and Results

In this chapter we describe the deployment of the laser localization system in a large scale assembly set up and present the experimental results. The objective is to evaluate how accurate our system is for positioning a robotic arm within an airplane wing.

### 6.1 Motivation & experimental set-up

Assembly operations in aircraft manufacturing are currently performed manually. Although aircrafts are small in lot size, numerous repetitive assembly operations have to be performed on a single aircraft. The conditions are often ergonomically challenging and these result in low productivity as well as frequent injuries. Thus, there is a need to shift from manual assembly to automated robotic assembly. The following wing-box assembly illustrates this.

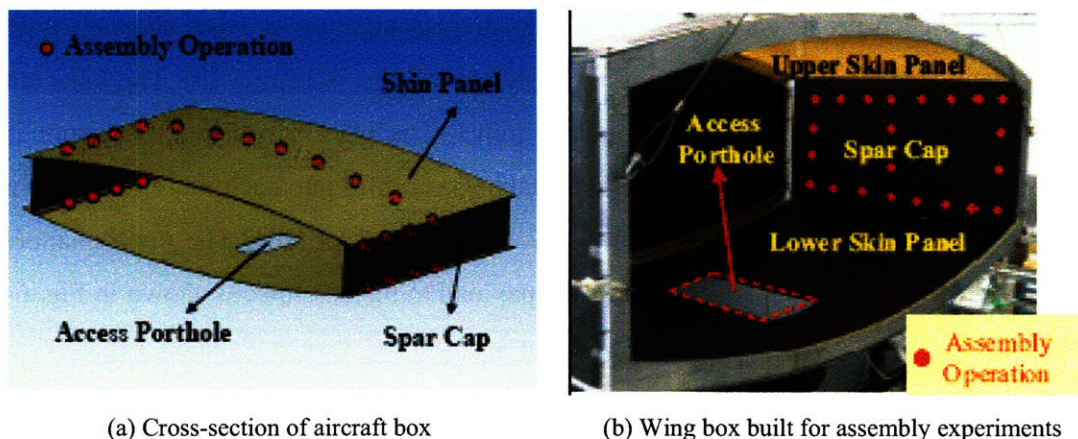


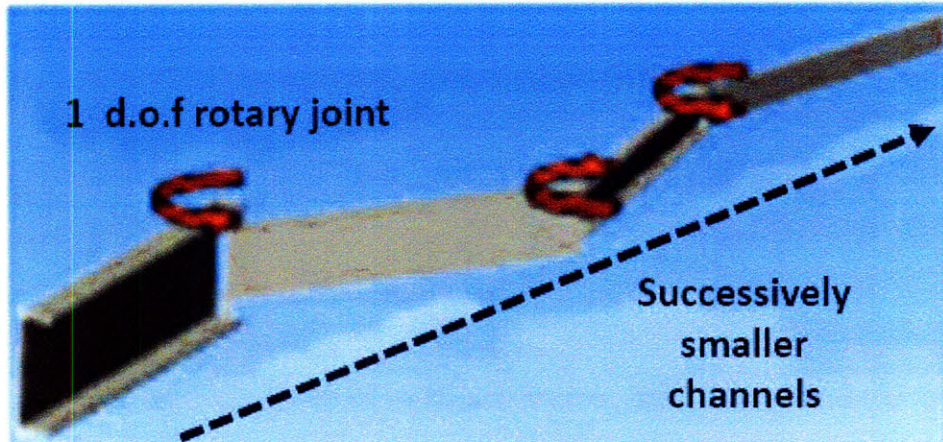
Figure. 6-1

Fig. 6-1 (a) shows a mock-up of the cross-section of an aircraft wing-box while Fig. 6-1 (b) contains a real life size dummy aircraft wing built at the d'Arbeloff Laboratory for Information Systems and Technology at MIT. Several assembly operations, such as burr-less drilling and fastener installations, have to be carried out inside the wing-box after the upper and lower skin panels are in place. The interior of the wing-box is accessible only through small portholes along its length. The portholes are roughly rectangular with dimensions of 45 cm by 23 cm. The wing-box also has a substantial span, which varies from 1 m to 3 m depending upon the size of the aircraft. The height of the wing-box varies from about 20 cm to 90 cm, once again depending upon the size of the aircraft. Presently, the assembly operations are carried out manually. A worker enters the wing-box through the small portholes and lies flat on the base, while carrying out the assembly operations. Evidently, the working conditions are ergonomically challenging.

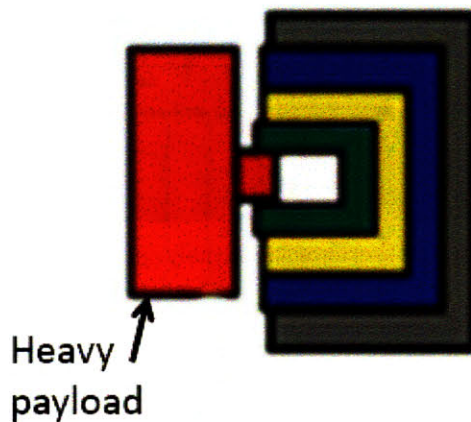
A robotic arm capable of performing such assembly operations should be compact enough to enter the wing-box through the small portholes. It should also be capable of subsequent reconfiguration, in order to perform the actual assembly operations at various locations inside the wing-box. There is also a relatively heavy payload attached to the tip of the arm hence it is indeed challenging to meet these diverse requirements in the design of a robot arm.

A deployable gravity assisted under actuated robotic arm with a serial linkage structure has been proposed in [23]. The links are essentially aluminum channels with successively smaller base and leg lengths, as shown in Figs. 6-2 (a) & (b). The links are connected by one degree of freedom rotary joints. The use of a channel structure is advantageous for a number of reasons. The channels can fold into each other resulting in an extremely compact structure during entry through the porthole. The open channel structure also facilitates the attachment of the payload to the last link, as shown in Fig. 6-2 (b).





(a) Structure of the robotic arm. The arm is made up of successively smaller aluminum links

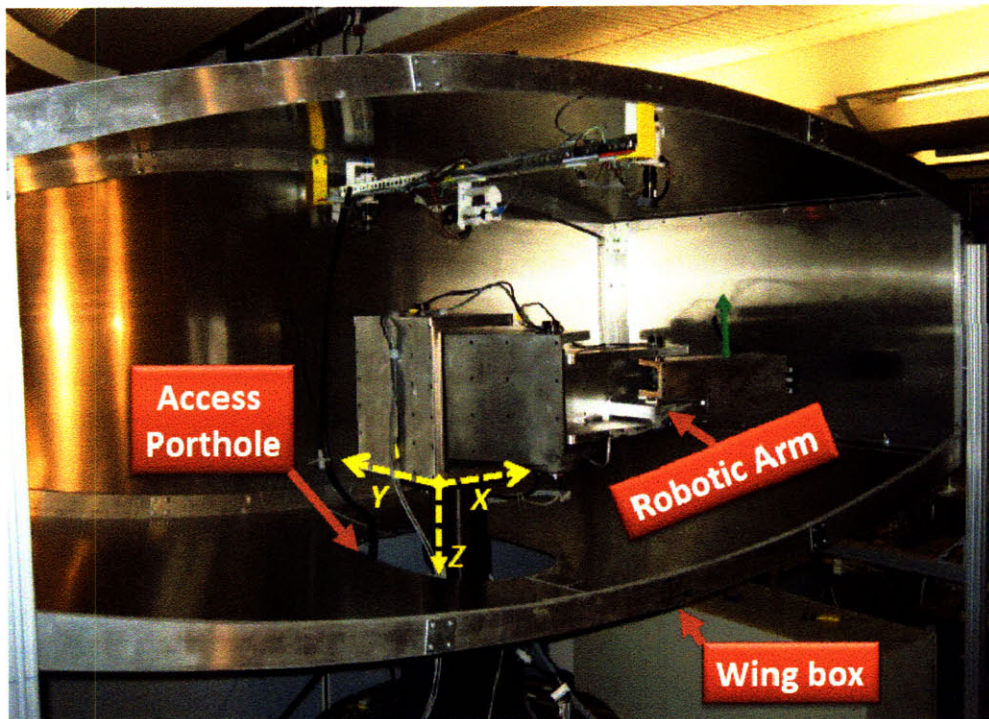


(b) Heavy payload attached to the final link of the robotic arm to perform assembly operations

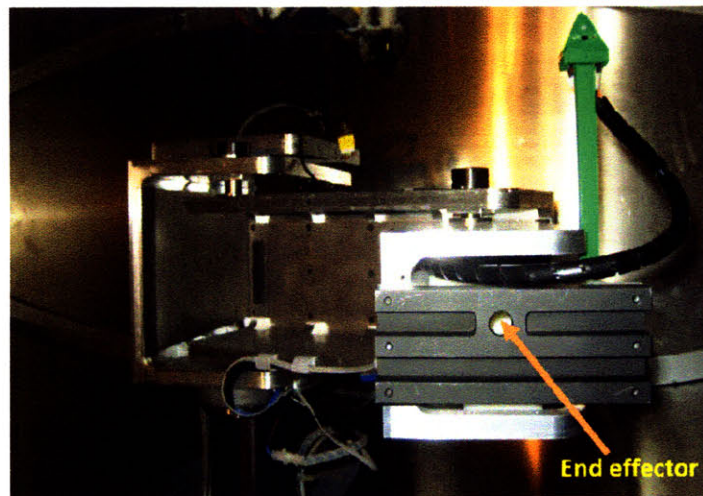
**Figure. 6-2**

The methodology requires a single actuator, which can be placed outside the wing-box and can be used in conjunction with simple locking mechanisms to reconfigure the serial linkage structure. The actuation scheme exploits gravitational and gyroscopic torques to rapidly deploy the manipulator arm inside the wing box. A picture showing the deployment of such a robotic arm inside the wing box is shown in Fig. 6-3.

A large end effector is mounted at the tip of the telescopic arm in order to perform the fitting and fastening operations inside the wing box (see Fig. 6-4). The heavy payload causes the telescopic arm to bend as it unlocks and extends individual links to reach out to the skin of the wing. The amount of bending is proportional of the weight of the payload and the magnitude of extension of the robotic arm.



**Figure. 6-3:** Robotic arm performing assembly operations inside the wing box.

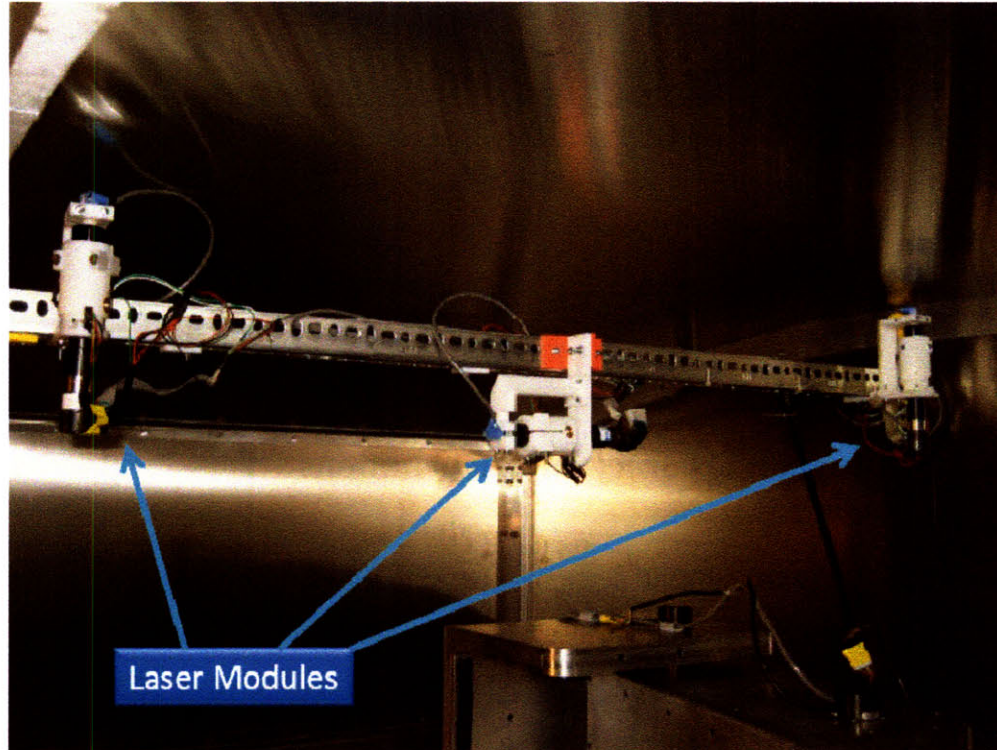


**Figure. 6-4:** An end effector, attached to the final link of the robotic arm, performs assembly/docking operations

In order to perform high-precision fitting, one has to continuously track and monitor the coordinates of the end effector inside the wing box. The encoders fitted to

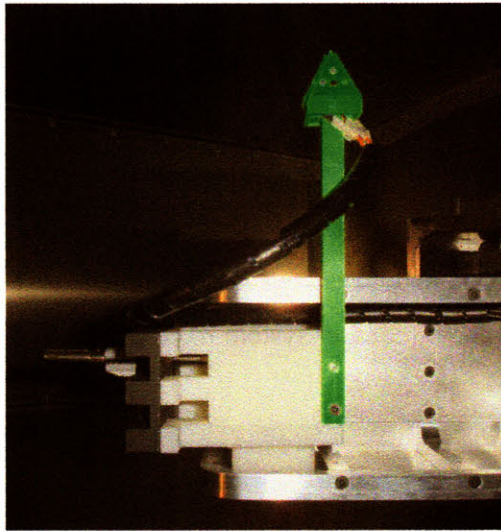
each of the unfolding links give us a rough estimate of the coordinates of the end effector, however, not to the precision levels that such operations demand. Moreover, by using the encoders or the hexapod on which the arm is mounted, one cannot find the amount of dropping of the tip of the arm which is crucial for high precision operations. Hence one needs an accurate indoor localization system to track the coordinates of the end effector thus enabling the robot to precisely dock on the target by compensating for the undesired effects.

In this chapter, we present the results of using the localization system we propose in this thesis to track the coordinates of the end effector and use feedback to guide the robotic arm for precise docking. As shown in Fig. 6-5, the three laser modules are mounted in a straight line lying in a vertical plane bisecting the wing box such that the midpoint of the three laser system lies directly above the center of the access porthole. The axes of measurement are chosen exactly as shown in Fig. 6-3 with the origin translated to the center of the three laser system fixed on the inner ceiling of the wing.



**Figure. 6-5:** Laser modules are mounted on the inner roof of the wing box in a straight line. This line lies in the center of the wing.

As shown in Fig. 6-6, the photodiode receiver module is attached to the end effector such that successive unfolding of the arm does not shadow the tetrahedral portion of the receiver module thus ensuring that there always exists a line of sight between each of the three lasers and at least one photodiode on the receiver. Once the coordinates of the centroid of the tetrahedral structure are obtained, the coordinates of the end-effector tip can be calculated by simple linear and trigonometrical transformations.



**Figure. 6-6.** Photodiode receiver mounted on the body of the end-effector. This point is chosen such that i) the receiver does not block the arm from unfolding, ii) receiver has direct line of sight & iii) the entire system (the arm & receiver module) fit through the access portal.

## 6.2 Experimental Results

### 6.2.1 Angular accuracy and distributions

The aim of this experiment was to determine the accuracy and spreads in measurement of angles by the laser system. In order to obtain the data, the robotic arm was locked in a specific position with the following true values for the angles:  $\theta_1 = 153.4349^\circ$ ,  $\theta_2 = 45^\circ$ ,  $\theta_3 = 153.4349^\circ$ . 500 readings (outputs of the laser system) each of the angles ( $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ), computed according to equation 5.3, were recorded with the arm held fixed at the above configuration. The histograms of recorded readings of the angles ( $\theta_1$ ,  $\theta_2$ ,

$\theta_3$ ) are shown in Fig. 6-7. The means and standard deviations of angles ( $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ) obtained from 500 readings recorded for this fixed position of the arm are as follows:

Angle	True values (in degrees)	Angles estimated by the laser system	
		Mean (in degrees)	Standard deviation (in degrees)
$\theta_1$	153.4349	153.4368	0.0205
$\theta_2$	45.00	45.0018	0.0203
$\theta_3$	153.4349	153.4335	0.0207

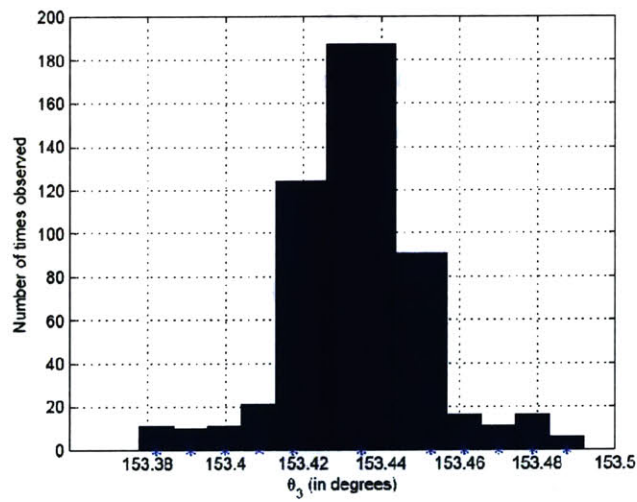
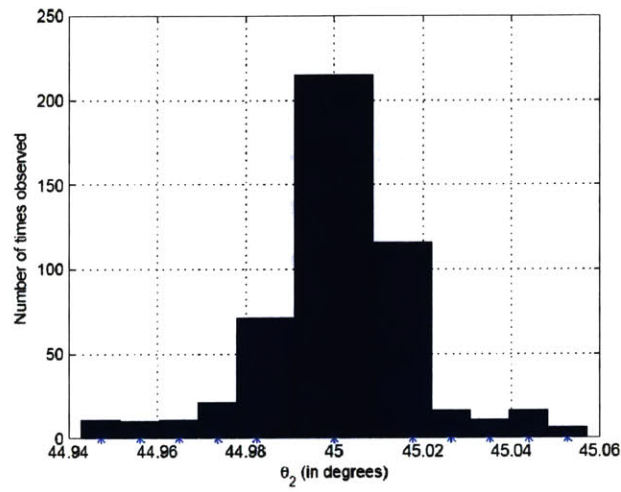
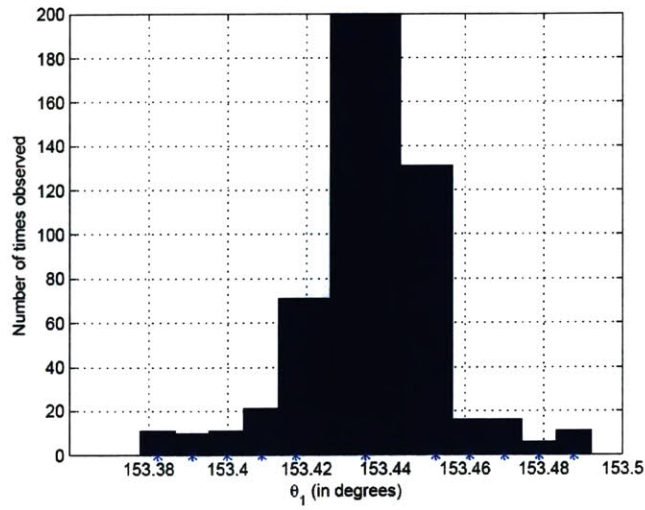
**Table. 6-1.** Means and standard deviations of angles ( $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ ) calculated from 500 observations of each. The robotic arm was held fixed during the experiment. Also tabulated are the true values of the angles

As demonstrated in the above table, the angle measurement capability of the system is highly accurate to within  $0.02^\circ$ . The accuracy levels can further be improved by reducing the amount of ambient AC light impinging on the photodiode module. As explained in chapter 4, the oscillating nature of the ambient light causes the edges of the photodiode pulse to move continuously hence effecting the timing measurement. Theoretically, in complete absence of ambient AC light, the system would be accurate to within less than  $0.01^\circ$  and this can be further reduced by increasing the least count of the encoders attached to the motors.

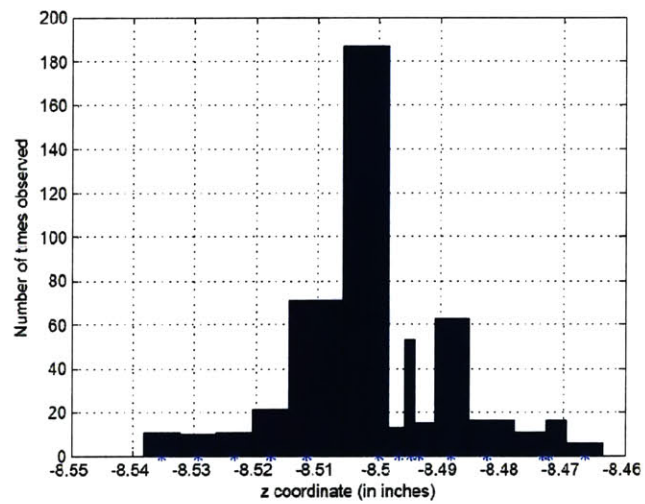
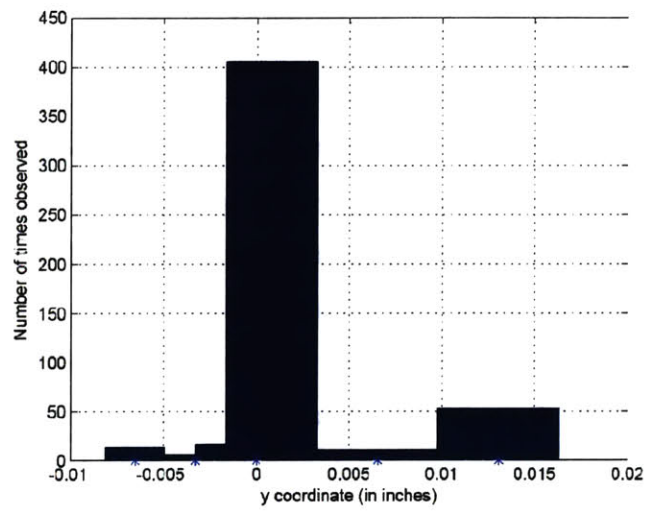
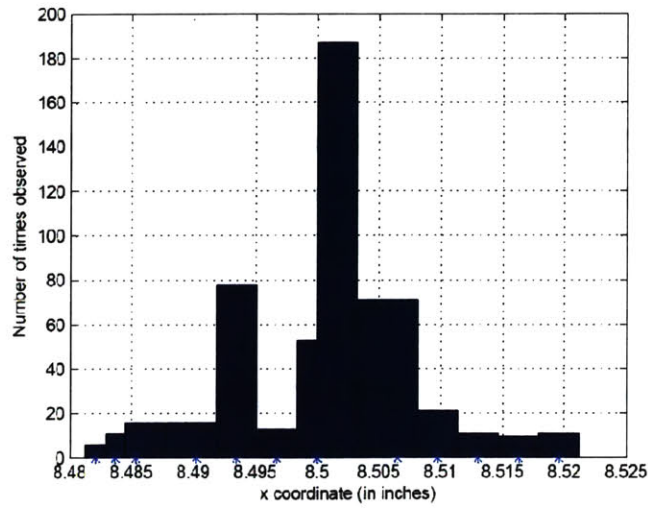
The corresponding distributions of the ( $x$ ,  $y$ ,  $z$ ) coordinates (calculated using the 500 angle readings recorded above and substituting them in equations 3.2 - 3.4) of the centroid of the photodiode receiver module are shown in Fig. 6-8 and tabulated in the following table:

Coordinate	Mean (in inches)	Standard deviation
$x$	8.4999	0.1854 mm
$y$	0.0012	0.1117 mm
$z$	-8.4995	0.3353 mm

**Table. 6-2.** Means and standard deviations of coordinates ( $x$ ,  $y$ ,  $z$ ) calculated from 500 observations of each. The robotic arm was held fixed during the experiment (see Fig. 6-8 for distributions)



**Figure. 6-7:** Angular distributions of  $(\theta_1, \theta_2, \theta_3)$ . 500 values of each were recorded with the robotic arm kept fixed a particular position. The means of the angles are taken as estimates of the true angles.



**Figure. 6-8:** Distributions of  $(x, y, z)$  coordinates. 500 values of each were recorded with the robotic arm kept fixed a particular position. The means of the coordinates are taken as estimates of the true coordinates.

As demonstrated in Table.6-2 and Figs. 6-8 (a), (b) and (c), the system is able to locate the receiver highly accurately with the error margins defined approximately by Gaussian distributions centered at their true values and with standard deviations of 0.1854, 0.1117 and 0.3353 mm in the  $x$ ,  $y$  and  $z$  directions respectively.

## 6.3 Theoretical error analysis

The localization technique presented in this thesis takes the measured values of angles  $\theta_1$ ,  $\theta_2$  &  $\theta_3$  as inputs and using formulas presented in equations 3.2, 3.3 and 3.4, calculates the coordinates  $(x, y, z)$ . Since measurements in angles  $\theta_i$  are not error proof (see Fig. 6-7) and have some inherent error (or uncertainty,  $\Delta\theta_i$ ), the values of coordinates calculated using equations 3.2-3.4 also have errors (due to propagation). In this section, we derive mathematical expressions to determine the theoretical values of uncertainties ( $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ ) in coordinates based on standard techniques in error analysis.

If a function  $f(a, b, c)$  is a dependent function of three independent parameters  $a$ ,  $b$  and  $c$ , then the error in  $f$  due to errors in parameters  $a$ ,  $b$  and  $c$  is given by:

$$\Delta f = \sqrt{\left(\frac{\partial f}{\partial a}\right)^2 (\Delta a)^2 + \left(\frac{\partial f}{\partial b}\right)^2 (\Delta b)^2 + \left(\frac{\partial f}{\partial c}\right)^2 (\Delta c)^2} \quad (6.1)$$

Thus, since (see equation 3.2),

$$x = \frac{-l}{(\cot \theta_1 + \cot \theta_3)}$$

we have,

$$\Delta x = \sqrt{\left(\frac{\partial x}{\partial \theta_1}\right)^2 (\Delta \theta_1)^2 + \left(\frac{\partial x}{\partial \theta_3}\right)^2 (\Delta \theta_3)^2} \quad (6.2)$$



Substituting (3.2) in (6.1), we get,

$$\Delta x = \sqrt{\left(\frac{l^2}{(\cot \theta_1 + \cot \theta_3)^4}\right) * (\operatorname{cosec}^4(\theta_1) \cdot (\Delta\theta_1)^2 + \operatorname{cosec}^4(\theta_3) \cdot (\Delta\theta_3)^2)}$$

similarly, since (see equation 3.3),

$$y = \frac{l}{2} + h \cot \theta_3$$

and,

$$\Delta y = \sqrt{\left(\frac{\partial y}{\partial \theta_1}\right)^2 (\Delta\theta_1)^2 + \left(\frac{\partial y}{\partial \theta_3}\right)^2 (\Delta\theta_3)^2}$$

we have,

$$\Delta y = \sqrt{\left(\frac{l^2}{(\cot \theta_1 + \cot \theta_3)^4}\right) * (\operatorname{cosec}^4(\theta_1) \cdot (\Delta\theta_1)^2 + \operatorname{cosec}^4(\theta_3) \cdot \cot^2 \theta_1 \cdot (\Delta\theta_3)^2)}$$

and lastly, since (see equation 3.4),

$$z = -h \cot \theta_2$$

and,

$$\Delta z = \sqrt{\left(\frac{\partial z}{\partial \theta_1}\right)^2 (\Delta\theta_1)^2 + \left(\frac{\partial z}{\partial \theta_2}\right)^2 (\Delta\theta_2)^2 + \left(\frac{\partial z}{\partial \theta_3}\right)^2 (\Delta\theta_3)^2}$$

we have,

$$\Delta z = \sqrt{\left(\frac{l^2}{(\cot \theta_1 + \cot \theta_3)^2}\right) * \left(\frac{\operatorname{cosec}^4(\theta_1) \cdot \cot^2 \theta_2 \cdot (\Delta \theta_1)^2}{(\cot \theta_1 + \cot \theta_3)^2} + \operatorname{cosec}^4(\theta_2) \cdot (\Delta \theta_2)^4 \dots \dots + \frac{\operatorname{cosec}^4(\theta_3) \cdot \cot^2 \theta_2 \cdot (\Delta \theta_3)^2}{(\cot \theta_1 + \cot \theta_3)^2}\right)}$$

From the data collected in section 6.2.1 and plotted in Fig. 6-7, for a particular fixed position of the receiver, we have the following values:

$$\theta_1 = 153.4368^\circ, \theta_2 = 45.0018^\circ, \theta_3 = 153.4335^\circ$$

*Means of distributions in Fig. 6-7*

And,

$$\Delta \theta_1 = 0.0205^\circ, \Delta \theta_2 = 0.0203^\circ, \Delta \theta_3 = 0.0207^\circ$$

*Standard deviations of distributions in Fig. 6-7*

Substituting these values in the equations derived above, we get:

$$\Delta x = 0.06858 \text{ mm},$$

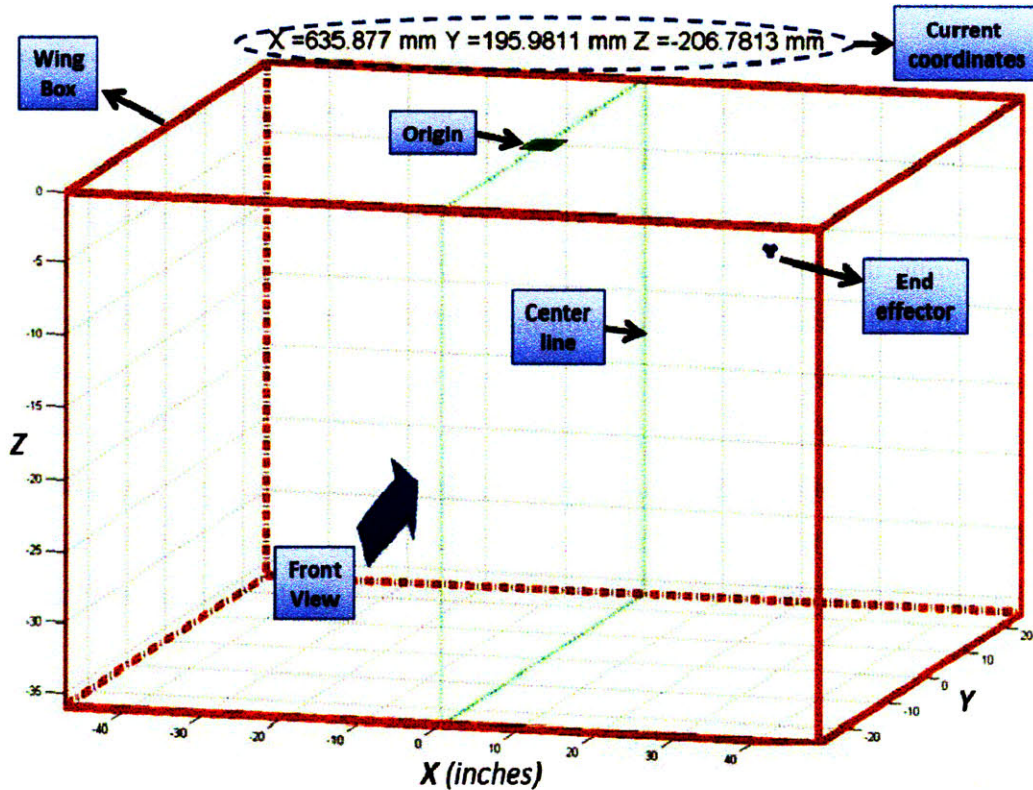
$$\Delta y = 0.1016 \text{ mm}, \text{ and}$$

$$\Delta z = 0.1828 \text{ mm}.$$

However, the uncertainty values obtained by practical experimentation (see Table. 6-2 and Fig. 6-8) were found to be  $\Delta x = 0.1854 \text{ mm}$ ,  $\Delta y = 0.1117 \text{ mm}$  and  $\Delta z = 0.3353 \text{ mm}$  which are a little higher than the values derived by theoretical analysis. This is because the formula for error propagation given in equation 6.1 is only a crude approximation but it is useful to get an estimate of the expected error in coordinate measurement given the uncertainty in angle measurement. One can reduce the uncertainty in angles  $\theta_i$  by a) reducing the amount of ambient AC light impinging on the photodiode

and *b*) increasing the resolution (or equivalently the number of pulses per revolution) of the encoders attached to the motors.

## 6.4 Real time tracking

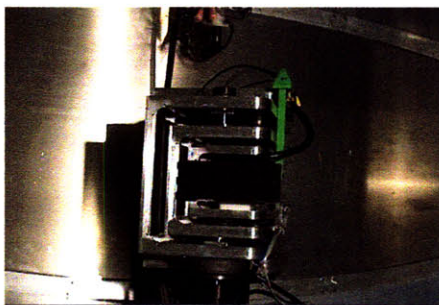


**Figure. 6-9:** 3D plotting tool for plotting the coordinates of the end effector. The red lines indicate the body of the wing and the solid blue marker indicates the current position of the end effector inside the wing

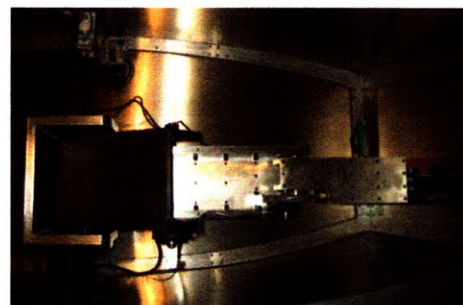
Real time tracking of the end-effector refers to the capability of the system to instantaneously plot the coordinates of the end-effector (as calculated from equations 3.2-3.4) in a virtual wing environment for easy visualization and tracking. A MATLAB script presented in Appendix B reads the angular data from the central control and processing unit, calculates the coordinates of the end effector as described in chapter 3 (equations 3.2 - 3.4) and plots the coordinates in a 3D visualization environment as shown in Fig. 6-9.

The red lines (solid and dotted) represent the walls of the wing-box and the  $x$ ,  $y$  and  $z$  axes are fixed according to the real dimensions of the wing-box.

Path tracing functionality is very useful in assembly operations using which the motion (or path taken) of the mobile robot can be monitored continuously and also analyzed offline in case of faulty behavior. The 3D plotting tool in Fig. 6-9 can also be used to trace the path taken by the end-effector while moving inside the wing-box. This can be done in MATLAB by using the command “hold on” before plotting the points in 3D (see the code in Appendix B). However, since the refresh rate of the localization system is 0.8 Hz, in order to get the correct value of the coordinates, the receiver must be held fixed for at least 1.15 seconds for each coordinate measurement. An experiment to test the path tracing capabilities of the localization system was also performed (see Fig. 6-11). During the experiment, the robotic arm started unfolding from the fully folded position to a fully extended position with the end-effector lying close to the skin of the wing in the end (see Fig. 6-10). This path was discretized into 27 points uniformly chosen over the entire path. During unwinding from the position shown in Fig. 6-10 (a), the robotic arm would stop at each of these points for approximately 2 seconds (this guarantees enough time for all the three lasers to update angle information) and the readings of the end-effector coordinates obtained from the localization system were recorded.



(a) Fully folded position



(b) Fully extended position

**Figure. 6-10:** Set-up for path tracing experiment. The arm unfolds from position (a) to (b) in a step-by-step fashion. Coordinates of its tip are tracked and recorded in order to trace the path it took.

The results of this experiment (plotted in Figs. 6-11 (a) and (b)) show a path very close to the actual path taken by the end effector during the experiment hence proving the

path tracing capabilities of the laser localization system. By using motors rotating at higher speeds (thus decreasing the refresh rate), one can obtain a continuous path tracing functionality which is very useful in the manufacturing and large-scale assembly industry.

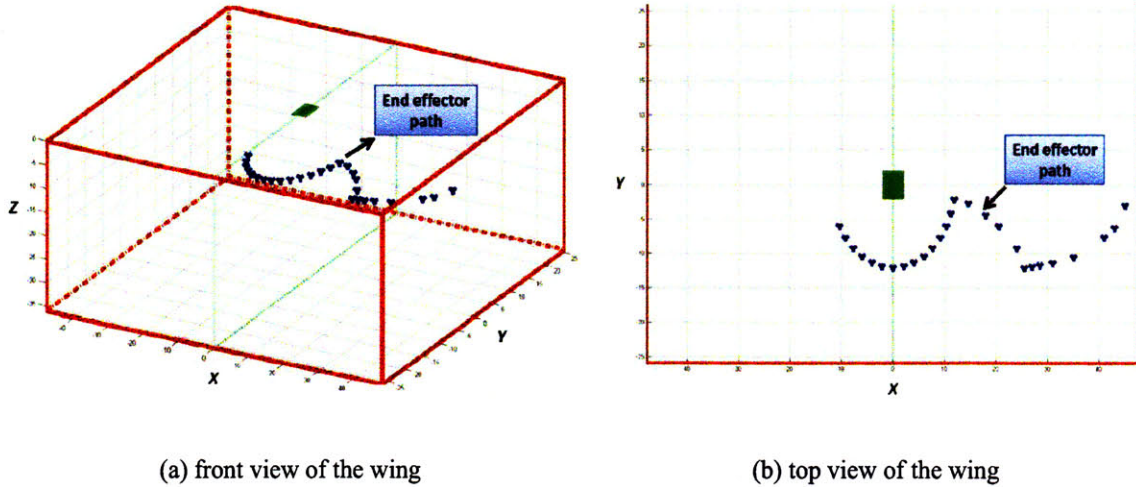


Figure. 6-11: Tracing of the path taken by the end-effector

## 6.5 Cross Validation

In this section, in order to test the accuracy of the localization system, we compare the coordinates of the end-effector as computed by three different methods:

- i) the laser localization system: coordinates are calculated using angles measured by the three laser modules and formulas derived in chapter 3.
- ii) using encoders: Each of the individual links in the robotic arm is equipped with encoders which measure their angular orientation with respect to their predecessor link. Hence, by knowing the lengths of each of the links and by using inverse kinematics, one can find the coordinates of the end effector. For example, for the configuration shown in Fig. 6-12, where  $\varphi_1 = 60^\circ$ ,  $\varphi_2 = 107^\circ$ ,  $\varphi_3 = 111^\circ$  and  $\varphi_4 = 154^\circ$ , the  $x$ ,  $y$  coordinates of the end-effector are calculated to be 41.042 and 0.189 inches respectively. The  $z$ -coordinate is determined by using the height of the hexapod on which the arm is mounted.

iii) ground truth: we choose a set of points (see Fig. 6-13) on the wing whose coordinates can be calculated by measuring their distances (using a tape) from the center of the wing-box. Hence we have an absolute location measurement of these points with respect to the ground reference frame.

In order to cross validate the coordinates as obtained by the above three methods, a docking operation is performed during which the robotic arm is guided in such a way so as to dock the end effector with the four rubber bumpers placed on the skin of the wing as shown in Fig. 6-13.

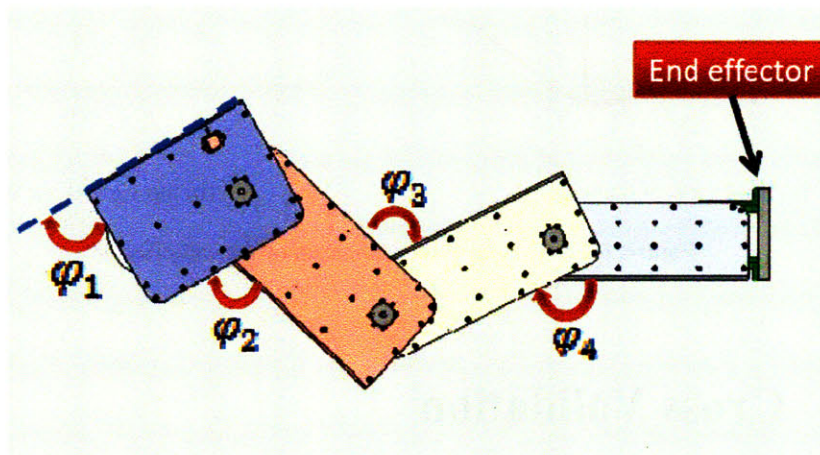


Figure. 6-12: Calculation of coordinates of end-effector from encoder readings

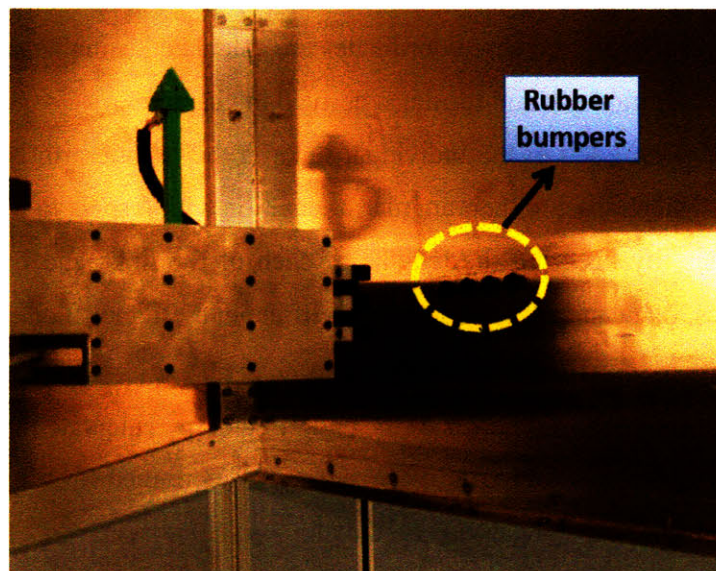


Figure. 6-13: Objects fitted on the skin of the wing on which the end-effector would dock precisely

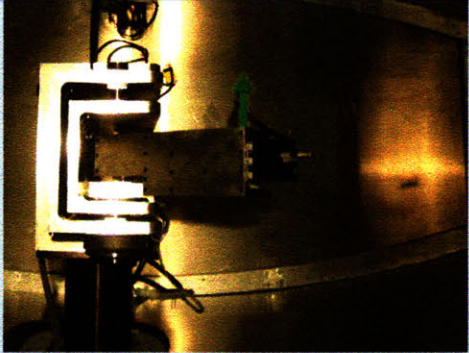
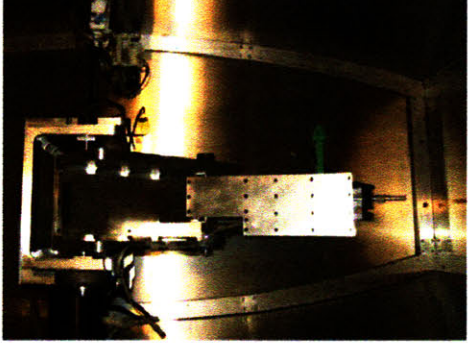
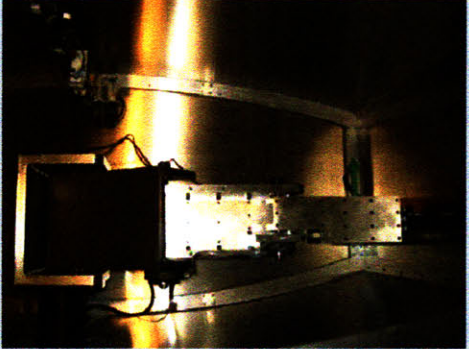
Table. 6-3 presents a comparative study of the coordinates of the end-effector as calculated by all the three methods.

Ground coordinates: Manual measurement ( $x, y, z$ )	Measurement with laser system (in inches)			Measurement with encoder readings (in inches)		
	$x$	$y$	$z$	$x$	$y$	$z$
(41.0, 0.2, -14.6)	41.044	0.188	-14.638	41.042	0.189	-14.505
(41.0, 1.0, -14.6)	41.044	0.975	-14.636	41.043	0.976	-14.505
(41.0, 1.8, -14.6)	41.045	1.765	-14.638	41.041	1.763	-14.505
(41.0, 2.5, -14.6)	41.044	2.553	-14.637	41.039	2.550	-14.505

**Table. 6-3.** Comparison of coordinates of the end effector as calculated by three methods: i) ground truth, ii) laser system readings & iii) encoder readings. Each of the 4 instances tabulated above correspond to the position of the end-effector when it is docking fully with the 4 rubber bumpers shown in Fig. 6-13

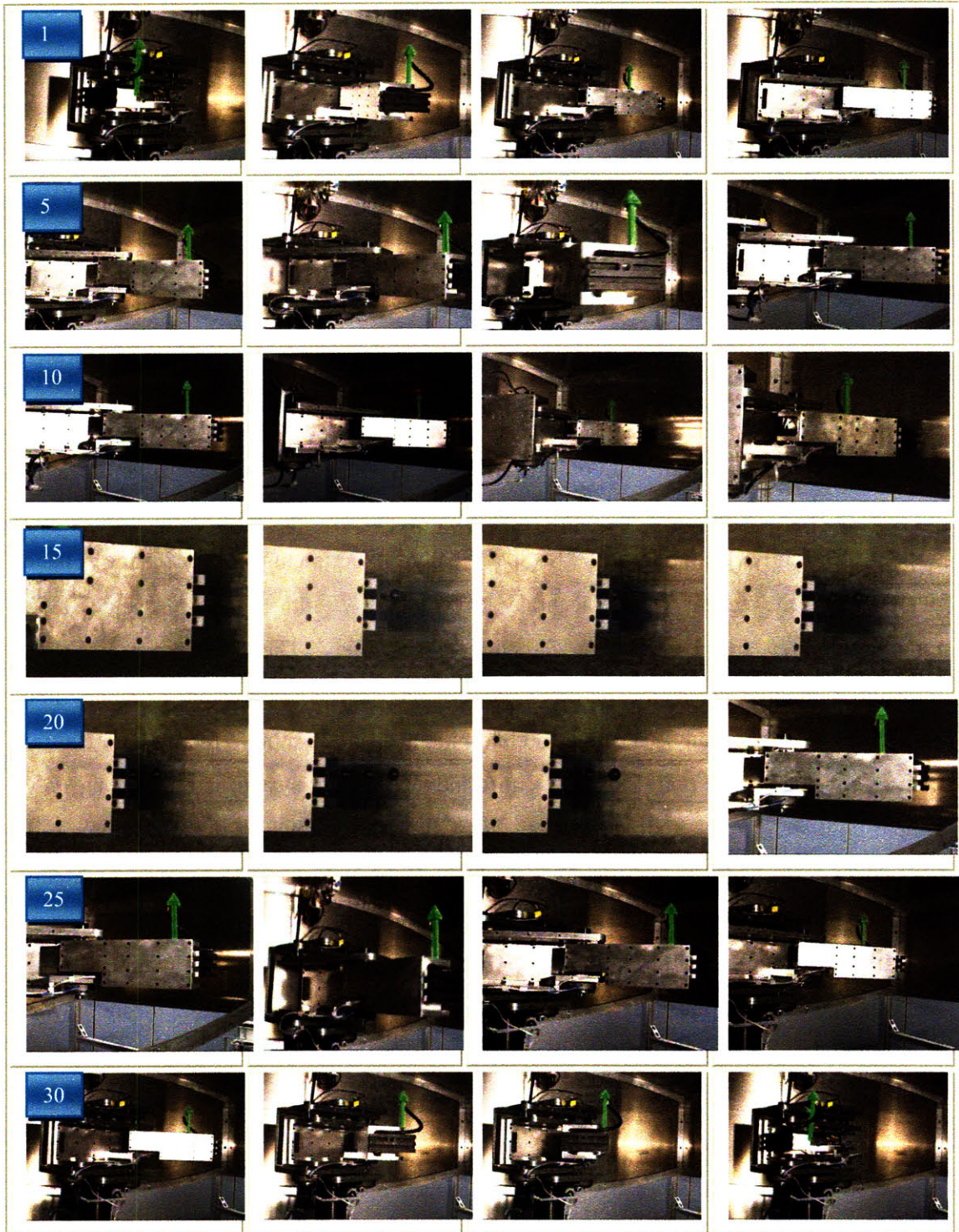
As observed from the table, the  $x$  and  $y$  coordinates obtained by all the three methods namely, using the proposed laser localization system, encoder readings and manual measurement, are in close proximity. However, the dropping of the extended arm in the  $z$ -direction cannot be found using the readings from the hexapod and the encoders. A more rigorous experiment demonstrating this drop in the tip of the arm due to its weight as it extends inside the wing is presented in Table 6-4. It was found that when fully extended, the tip of the arm dropped by 3.39 mm in the  $z$  direction.

Using the feedback from the laser localization system, a docking operation on the four bumpers shown in Fig. 6-13 was successfully performed and demonstrated in Fig. 6-14. The robot control system uses the encoder readings to plan and move the arm to near the vicinity of each of the bumpers. Once it is in the proximity of the bumpers, it reads the coordinates calculated by the laser system through an Ethernet connection (see Fig. 5-1) and accordingly adjusts the height of the hexapod in order to compensate for the drop of the tip in the  $z$ -direction.

Arm extension		z coordinate of the tip (receiver) obtained by laser system	z coordinate of the tip obtained by hexapod readings
<b>Initial position</b>		-368.43 mm	-368.43 mm
<b>Half-extended position</b>		-369.41 mm	-368.43 mm
<b>Fully extended position</b>		-371.82 mm	-368.43 mm

**Table. 6-4.** This table compares the readings of the laser system with that of the hexapod on which the arm is mounted. Due to the weight of the links, a drop in the tip of the arm is observed. A drop of 3.39 mm in the tip (in the z-direction) is detected by the laser system. This phenomenon is not detected using hexapod readings.





**Figure. 6-14:** Robotic arm performing precise docking operations using feedback from the localization system. This figure shows frames captured from a video of the docking operation.

# Chapter 7

## Conclusions

In this thesis we have proposed, designed, built and successfully demonstrated a novel localization system to determine the position of a mobile robot in an indoor environment. The system advances the state of the art by achieving accuracy levels in the sub-millimeter range which has not been achieved by any of the commercial localization systems currently available [1, 4]. Specifically, experiments in localization have demonstrated the system to have accuracy levels of 0.1778 mm, 0.1016 mm and 0.3352 mm in the  $x$ ,  $y$  and  $z$  coordinate directions respectively. The system consists of a transmitter module consisting of three rotating line lasers mounted in a straight line configuration, a photodiode receiver module mounted on the mobile robot and a central control and processing unit. Since the entire indoor tracking system is build using only simple light sources, photodiodes and an FPGA board, the total cost of development is well under \$200 which is significantly less than all the commercial localization systems currently available [4]. Coordinates of the robot are determined using triangulation algorithms that are based on the precise measurement of the angular orientations of the receiver module with respect to the transmitters. A unique tetrahedral design, with photodiodes mounted on each of its four faces in order to ensure a  $360^\circ$  field of view, is used for the receiver and a center of mass equivalent algorithm is used to determine the coordinates of its centroid.

We successfully deployed and used the localization system in an aircraft assembly set-up. Aircraft industry has long lagged behind in the use of automated robots and even today bulk of the assembly is done by human workers. Since most of the aircraft body is accessible only through small portholes, a compact reconfigurable

telescopic arm has been designed and built in [23]. However, due to the heavy payload, the robotic arm drops down in the  $z$ -direction as the arm extends in order to perform assembly operations on the skin of the wing. Accurate tracking of the coordinates of the end effector is quintessential to the problem of performing any high precision fitting or assembly tasks inside the wing. By using the localization system proposed in this thesis, we have demonstrated its capability in accurately finding the drop in the robotic arm's tip as it extends inside the wing-box (experiments indicate an average drop of 3.4 mm on full extension of the arm). Using the feedback from the proposed localization system, the robotic arm was successfully guided to perform a high-precision docking operation inside the wing.

Note: All the hardware and software files used in the implementation of the localization system presented in this thesis can be accessed online at: <http://groups.csail.mit.edu/drl/wiki/index.php/lasersystem>

## 7.1 Lessons learned

During the course of the project, we learned a lot of valuable lessons. Practical experiments in robotics are inherently complicated and involve finding smart engineering solutions to problems which never seem existent till we proceed to real-life experimentation. Our initial solution to the localization problem involved using only two laser transmitter modules, a single photodiode receiver and a Philips LPC 2148 microcontroller board. Omni-directionality of the photodiode receiver was achieved by using a reflective metallic sphere mounted over the photodiode receiver. Duration of shining of the laser beam on the photodiode was used to calculate the perpendicular distance between the photodiode and the transmitters (see Fig. 3-7). Although, the initial testing of the system seemed promising, upon actually deploying the system inside the wing box, we realized that the distance measurements were highly sensitive to oscillations of the signals due to ambient light especially at distances above 500 mm (see Fig. 3-7) which were not easily discernable using timing information. We also had huge areas inside the wing box where the receiver module would be completely shadowed by the unfolding links of the arm hence making tracking in these regions impossible.

However, through these experiments, not only did we gain valuable experience in embedded C programming, PCB design and SolidWorks but also got inspired to think of a more robust and accurate solution to the current problem while taking into account all the practical challenges we faced during our first stage.

Our experiences with LPC microcontrollers have been mixed. Although, the LPC microcontrollers are easy to program and debug, they are not flexible and cannot be easily expanded to accommodate new digital circuitry. For example, during the implementation of the current system, due to the use of four photodiodes instead of just one in the initial system, we needed to have four 32-bit counters with 2 capture pins each which was not possible to achieve using LPC microcontrollers. Hence, we switched to use Xilinx FPGA boards for integrated circuit development. Xilinx FPGA boards (manufactured by Basys) are highly flexible, fast, inexpensive, easy to program, have large set of I/O ports onboard and can easily be extended to add complex new circuitry.

For our initial implementation, we had used Matlab as the sole software to read the data from the serial port of the computer and plot the 3D visualizations. This implementation turned out to be extremely slow and inefficient. Later, we used Java to implement a multi-threading architecture which would continuously poll the serial port for new data. Our Matlab code then simply used these Java objects to get the data and plot the data in 3D. This resulted in a remarkable improvement in the speed of the software.

We also realized that calibration of the localization system is a big challenge. If high levels of accuracy are to be obtained, it is extremely important to know the precise values of misalignments between the center of the wing box and the center of the laser system, between the centroid of the receiver module and the end effector, any effects due to tilting of the wing, etc, and later compensate for them in software. Before using the localization system, it is crucial to precisely measure the exact coordinates of the centroid of the receiver module when the end effector is accurately performing its fitting or assembly operations and only then use the readings from the laser system as feedback to control the motion of the arm.

In summary, the following are the sensitivities and limitations of the laser localization system proposed in this thesis:

### 7.1.1 Sensitivities

- a) The current localization system is sensitive to ambient AC light. Because of the constant threshold comparator which used to produce a square pulse whenever the laser beam shines on the photodiode (see Appendix A), the oscillating nature of the ambient light causes spurious shifts in the edges hence effecting the angle measurements. A few ways of minimizing this effect are: i) working with DC light sources in the environment and ii) using advanced filtering techniques (adaptive filters) to filter out the oscillating signals.
- b) The system is sensitive to misalignments. Possible sources of misalignments are i) between the center of the wing and center of three laser system (this can be tackled by pre-designing the wing with holes or slots placed accurately where the lasers could be fitted), ii) the tilt of the wing-box and the laser mounting rod (see Fig. 6-5), and iii) finding the center of the three laser system which would act as the center of the coordinate system in which the coordinates (see equations 3.2 - 3.4) would be calculated. This effect can be minimized by manufacturing compact laser modules whose centers can be easily calculated. Another technique that can be used to tackle this issue of misalignments is to manually record the readings (coordinates) of the receiver when the end-effector is aligned perfectly with the target job (this has to be done before the actually fitting) and then during the experiment, continuously use the laser system as feedback to guide the arm such that the final coordinates are the same as the recorded ones.
- c) The system is sensitive to high speed movements of the arm. Due to the rotation speed of the motors, the system has a refresh rate of 0.8 Hz. This means that in order to get correct coordinates, the arm has to be stationary for at least 1.15 seconds which would allow all the three lasers to update their angular measurements. This problem can be overcome by increasing the speed of the motors or by using the system readings only when performing highly accurate

jobs where speedy motion is not a priority (for example, while performing assembly jobs on the skin of the wing).

### 7.1.2 Limitations

- a) The laser localization system presented in this thesis cannot be used in outdoor environments where there is bright ambient light (such as sunlight) as this would cause the photodiodes to saturate.
- b) The localization system provides us only with 3D coordinates  $x$ ,  $y$  and  $z$  and additional information regarding the geometry of the robot and its environment is necessary to find its orientation. For example, for the assembly experiments presented in this thesis, we knew apriori the orientation of the robotic arm when it is close to the skin of the wings (we knew that the orientation of the end effector would be parallel to the surface of the skin or equivalently parallel to the  $y$ - $z$  plane), and hence we were able to calculate the coordinates of the end-effector using simple linear transformations.
- c) Line-of-sight of the receiver with the laser transmitters is important for the system to work. Therefore, given an unknown environment, multiple laser transmitters must first be placed so that line of sight is maintained at all times during robot's motion. Transmitter placement and identification techniques we describe in section 3.2 and 3.3 can be used in such scenarios.
- d) *Portability*: For the system to be used in a new application, it must satisfy the following criterion: i) we must have apriori knowledge of the environment, the obstacles, target locations etc. Therefore the system can be used for localization in applications such as large scale assembly, robot navigation through known environments, etc but cannot be used for map building, ii) there must be a direct line of sight between the lasers and the receiver and the receiver must be able to detect the laser beam (cannot be used in outdoor sunlight) and iii) there must not be any bending of laser light rays as this affects the measurement of angles. Hence it cannot be used in mediums other than air or vacuum (for example in water).

## 7.2 Future Work

The laser localization system presented in this thesis does not enable us to find the orientation of the mobile robot. The orientation of the robot can be found out by knowing which of the photodiodes in the receiver have been exposed to the laser. However, if such a technique is used in the current set-up, we get a resolution of only  $\frac{360^\circ}{4} = 90^\circ$ . Instead, if one can use a cylindrical photodiode receiver with photodiode sensors all along its curved surface, a much higher resolution for orientation can be obtained.

Oscillations due to ambient light degrade the accuracy of the system considerably. Designing adaptive filters which filter ambient light frequencies and yet do not significantly attenuate the photodiode pulses is a challenging task and needs further investigation.

The slip rings are mounted such that the structure that holds the slip rings over the motors (see Fig. 4-4) shadows one half of the wing. Therefore, if assembly operations on the other side of the wing have to be tracked, a slip ring holder has to be redesigned to provide a  $360^\circ$  field of view for the laser.

Though we have very briefly discussed possible solutions in sections 3.2 and 3.3, techniques to find the optimal transmitter placement to provide coverage in a unknown 3D environment need further investigation and testing. One can think of integrating the laser localization system with one of the commercially available 3D laser scanning systems [24] using which one can first create a 3D map of the environment and then using algorithms similar to ones presented in sections 3.2 and 3.3, choose points to mount transmitters to ensure complete coverage of the environment. Thus this system can be extended and used for navigation of robots in an unknown 3D environment with no a priori information about the geometry of the surroundings.





# Appendix A

## Electrical Schematics

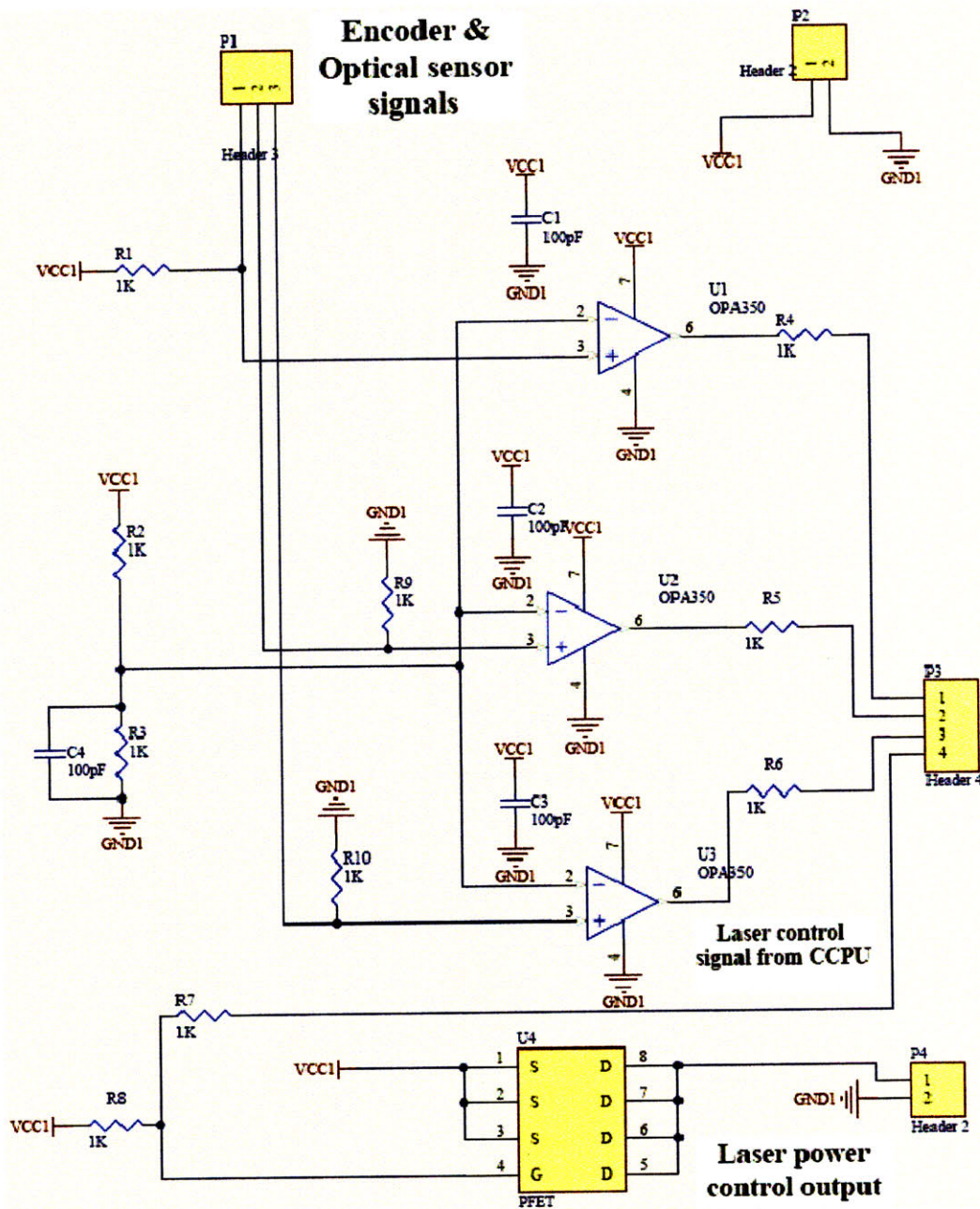
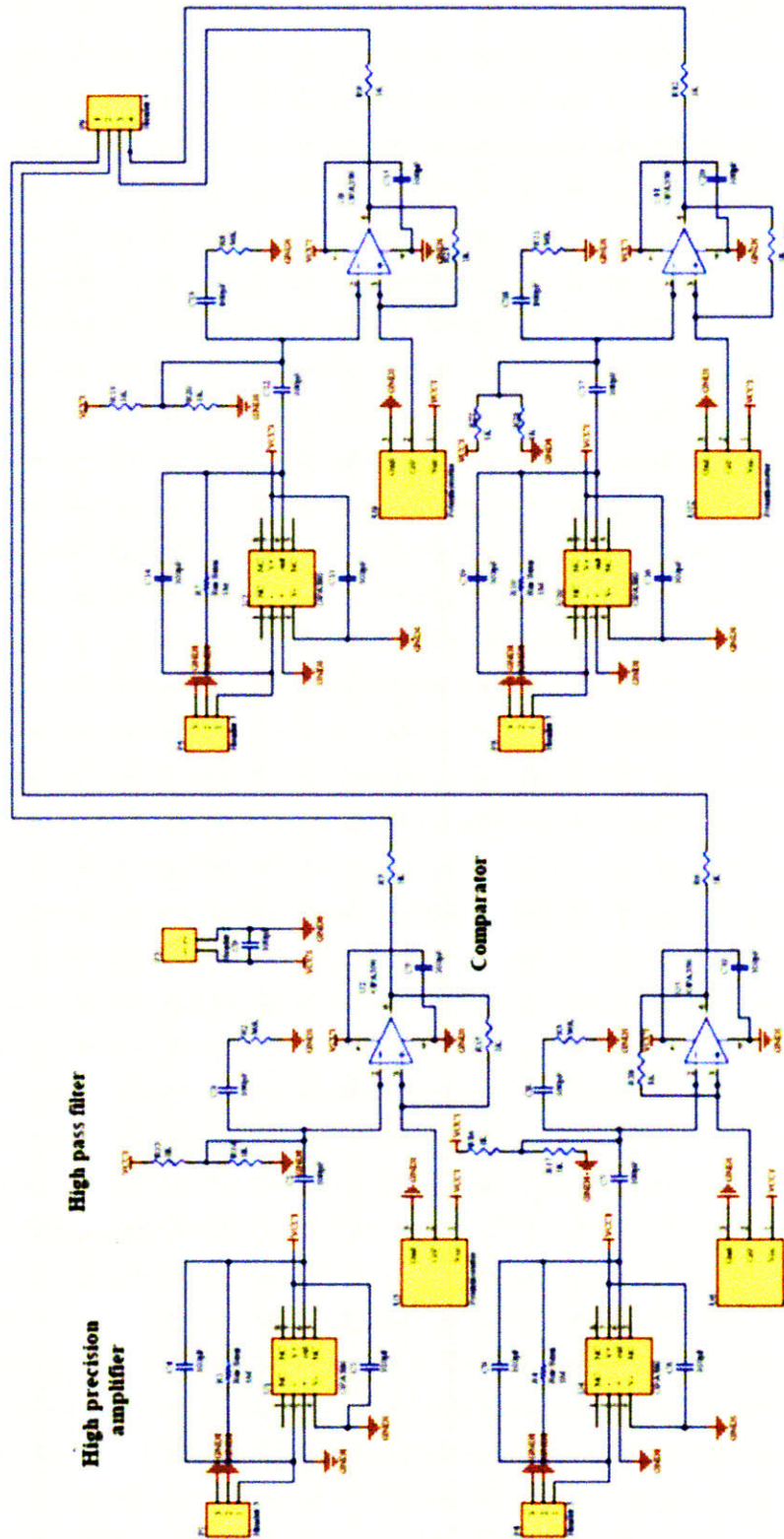


Figure. A-1: This circuit performs two tasks: i) transmit signals from encoders and optical sensors to the CCPU & ii) controls the power supply to lasers depending on the signal from CCPU



**Figure. A-2:** Photodiode signal amplifier circuit. This circuit consists of 4 high-precision amplifiers, 4 highpass filters and 4 comparators (one for each photodiode in the receiver module).

# Appendix B

## Software

### B-1. FPGA code for central control and processing unit

```
module main(EN, CLK, PHOTO, ENCODER1, ENCODER2, BEAM_BREAK, LASER, SERIAL_OUT, LED);
  parameter PHOTOS = 4;
  parameter LASERS = 3;
  parameter COUNTER_BYTES = 4;
  parameter BYTE_SIZE = 8;
  parameter INPUT_CLK = 100000000;
  parameter STATE_SPEED = 1000000;
  parameter SERIAL_SPEED = 115200;

  localparam STATE_DIVIDER = INPUT_CLK / STATE_SPEED;
  localparam SERIAL_DIVIDER = INPUT_CLK / SERIAL_SPEED;
  localparam COUNTER_SIZE = COUNTER_BYTES*BYTE_SIZE;
  localparam DATA_SIZE = COUNTER_SIZE*PHOTOS*2;

  input          CLK, EN;
  input [PHOTOS-1:0] PHOTO;
  input [LASERS-1:0] ENCODER1, ENCODER2;
  input [LASERS-1:0] BEAM_BREAK;
  output [LASERS-1:0] LASER;
  output          SERIAL_OUT;
  output [7:0]      LED;

  localparam STATE_WAIT_EN = 0;
  localparam STATE_PHOTO = 1;
  localparam STATE_SERIAL = 2;

  reg [2:0] state = STATE_WAIT_EN;
  reg [LASERS-1:0] current_laser = 1;

  wire [LASERS-1:0] ENCODERS = ENCODER1 ^ ENCODER2;
  wire current_encoder = | (ENCODERS & current_laser);
  wire current_beam_break = | (BEAM_BREAK & current_laser);
  wire [DATA_SIZE-1:0] DATA;
  wire cc_done, state_clk;
  wire cc_en = state == STATE_PHOTO;
  wire serial_en = state == STATE_SERIAL;
  wire serial_done, serial_clk;

  assign LASER = ~current_laser;

  assign LED[1:0] = current_laser;
  assign LED[4:2] = state;
  assign LED[5] = serial_done;
  assign LED[6] = cc_done;
  assign LED[7] = current_beam_break;

  divider #(.DIVIDER(STATE_DIVIDER)) d_sc(CLK, state_clk);

  control_capture #(.COUNTER_SIZE(COUNTER_SIZE), .PHOTOS(PHOTOS))
    cc (.SCLK(state_clk), .ENCODER(current_encoder), .BEAM_BREAK(current_beam_break),
      .EN(cc_en), .PHOTO(PHOTO), .DONE(cc_done),
      .OUT_RISE(DATA[COUNTER_SIZE*PHOTOS-1:0]),
      .OUT_FALL(DATA[COUNTER_SIZE*PHOTOS*2-1:COUNTER_SIZE*PHOTOS]));
endmodule
```

```

divider #(.DIVIDER(SERIAL_DIVIDER)) d_se(CLK, serial_clk);

wire [7:0] current_id = current_laser + 8'd97;

serial_out #(.DATA_BYTES(3+DATA_SIZE/BYTE_SIZE), .BYTE_SIZE(BYTE_SIZE))
  so (.CLK(serial_clk), .EN(serial_en), .DATA({DATA, current_id, 8'd10, 8'd13}),
    .DONE(serial_done), .OUT(SERIAL_OUT));

always @(posedge CLK) if(EN) begin
  case(state)
    STATE_WAIT_EN: begin
      state <= STATE_PHOTO;
      current_laser <= 1;
    end
    STATE_PHOTO: if(cc_done) begin
      current_laser <= {current_laser[LASERS-2:0], current_laser[LASERS-1] };
      state <= STATE_SERIAL;
    end
    STATE_SERIAL: if(serial_done) begin
      state <= STATE_PHOTO;
    end
  endcase
end else begin
  state <= STATE_WAIT_EN;
end

endmodule

module control_capture(SCLK, ENCODER, BEAM_BREAK, EN, PHOTO, DONE, OUT_RISE, OUT_FALL);
  parameter COUNTER_SIZE = 32;
  parameter PHOTOS = 4;

  input SCLK, ENCODER, BEAM_BREAK, EN;
  input [PHOTOS-1:0] PHOTO;
  output DONE;
  output [COUNTER_SIZE*PHOTOS-1:0] OUT_RISE, OUT_FALL;

  localparam ST_WAIT_HIGH1 = 0;
  localparam ST_WAIT_LOW1 = 1;
  localparam ST_WAIT_HIGH2 = 2;
  localparam ST_WAIT_LOW2 = 3;
  localparam ST_DONE = 4;
  localparam ST_WAIT_HIGH0 = 5;
  localparam ST_WAIT_LOW0 = 6;

  reg [2:0] state = ST_WAIT_HIGH1;
  wire en_capture = (state == ST_WAIT_HIGH2) | (state == ST_WAIT_LOW2);
  assign DONE = (state == ST_DONE);
  reg [COUNTER_SIZE-1:0] counter;

  genvar i;
  generate
    for(i=0; i<PHOTOS; i=i+1) begin: ph
      photo_capture #(.COUNTER_SIZE(COUNTER_SIZE))
        pc( .CLK(ENCODER), .COUNTER(counter),
          .EN(en_capture), .PHOTO(PHOTO[i]),
          .OUT_RISE(OUT_RISE[(i+1)*COUNTER_SIZE-1:i*COUNTER_SIZE]),
          .OUT_FALL(OUT_FALL[(i+1)*COUNTER_SIZE-1:i*COUNTER_SIZE]));
    end
  endgenerate

  always @(posedge SCLK) begin
    if(EN) begin
      case(state)

```

```

        ST_WAIT_HIGH1: if(BEAM_BREAK) state <= ST_WAIT_LOW1;
        ST_WAIT_LOW1:  if(~BEAM_BREAK) state <= ST_WAIT_HIGH2;
        ST_WAIT_HIGH2: if(BEAM_BREAK) state <= ST_WAIT_LOW2;
        ST_WAIT_LOW2:  if(~BEAM_BREAK) state <= ST_DONE;
    endcase

    end else begin
        state <= ST_WAIT_HIGH1;
    end
end

always @(posedge ENCODER)
    if(en_capture)
        counter <= counter + 1;
    else
        counter <= 0;

endmodule

module photo_capture(CLK, COUNTER, EN, PHOTO, OUT_RISE, OUT_FALL);
    parameter COUNTER_SIZE = 32;

    input CLK, EN, PHOTO;
    input [COUNTER_SIZE-1:0] COUNTER;
    output [COUNTER_SIZE-1:0] OUT_FALL, OUT_RISE;

    reg [COUNTER_SIZE-1:0] OUT_FALL, OUT_RISE;
    reg state = 1;
    reg last_photo = 0;

    always @(posedge CLK) begin
        if(EN) begin
            if(state) begin
                state <= 0;
                OUT_FALL <= 0;
                OUT_RISE <= 0;
            end else begin
                if(~last_photo && PHOTO)
                    OUT_RISE <= COUNTER;
                if(last_photo && ~PHOTO)
                    OUT_FALL <= COUNTER;
            end
        end else begin
            state <= 1;
        end
        last_photo <= PHOTO;
    end
endmodule

module serial_out(CLK, EN, DATA, OUT, DONE);
    parameter DATA_BYTES = 2;
    parameter BYTE_SIZE = 8;
    localparam BIT_COUNTER_SIZE = 12; // log of DATA_BITS

    localparam DATA_SIZE = DATA_BYTES*BYTE_SIZE;
    localparam COOKED_SIZE = DATA_BYTES*(BYTE_SIZE+2);

    localparam STATE_WAIT_EN = 0;
    localparam STATE_TRANSMIT = 1;
    localparam STATE_DONE = 2;

    input CLK, EN;
    input [DATA_SIZE-1:0] DATA;
    output OUT, DONE;

    integer i, j;

```

```

reg [COOKED_SIZE-1:0]    cooked_data;
reg [BIT_COUNTER_SIZE-1:0] counter;
reg [1:0]                state = STATE_WAIT_EN;

assign DONE = state == STATE_DONE;
assign OUT = (state != STATE_TRANSMIT) || cooked_data[0];

always @(posedge CLK)
if(EN) begin
  case(state)
    STATE_WAIT_EN: begin
      state <= STATE_TRANSMIT;
      counter <= 0;
      for(i=0;i<DATA_BYTES;i=i+1) begin
        cooked_data[i*(BYTE_SIZE+2)] <= 0;
        cooked_data[i*(BYTE_SIZE+2)+BYTE_SIZE+1] <= 1;
        for(j=0;j<BYTE_SIZE;j=j+1)
          cooked_data[i*(BYTE_SIZE+2)+ 1 + j] <= DATA[i*BYTE_SIZE + j];
      end
    end
    STATE_TRANSMIT: begin
      if(counter == COOKED_SIZE-2) begin
        state <= STATE_DONE;
      end
      counter <= counter + 1;
      cooked_data <= cooked_data >> 1;
    end
    default: begin
      counter <= counter + 1;
      cooked_data <= cooked_data >> 1;
    end
  endcase
end else begin
  state <= STATE_WAIT_EN;
  cooked_data <= cooked_data >> 1;
  counter <= 0;
end
endmodule

module divider(CLK, OUT);
parameter DIVIDER = 434;
parameter COUNTER_SIZE = 16;

input CLK;
output OUT;

reg [COUNTER_SIZE-1:0] counter = 0;
reg OUT ;

always @(posedge CLK) begin
  if(counter == DIVIDER-1) begin
    counter <= 0;
    OUT <= 1;
  end else begin
    counter <= counter+1;
    if(counter == DIVIDER/2-1)
      OUT <= 0;
  end
end
endmodule

```

## B-2. Java code for reading data from CCPU to computer

```
package lasersreading;

import javax.comm.*;
import java.io.*;

public class LasersReader extends Thread (

    int lasersCount, photoCount, countersSize;
    int readings[][];
    boolean running;
    SerialPort com;
    OutputStream out;
    InputStream in;

    public LasersReader(String serialport, int speed, int lasersCount, int photoCount, int
        countersSize)
        throws Exception{
        com = (SerialPort) CommPortIdentifier.getPortIdentifier(serialport).open("test",
            2000);
        com.setSerialPortParams(speed,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);

        out = com.getOutputStream();
        in = com.getInputStream();
        this.lasersCount = lasersCount;
        this.photoCount = photoCount;
        this.countersSize = countersSize;
        if(countersSize>4) throw new RuntimeException("Counter size more than int (4
            bytes)");
        readings = new int[lasersCount][photoCount*2];
    }

    public void stopReading() throws Exception{
        in.close();
        out.close();
        com.close();
        running = false;
    }
}
```

```

public void startReading(){
    running = true;
    start();
}

public void run(){
    try{
        while(running){
            while(in.read() != 0x0D); // System.out.println("no");
            if(in.read() != 0x0A) continue;
            int readLaser = in.read() - 0x61;
            int laser = 1;
            //System.out.println("beginning");
            for(int i=0;i<lasersCount;i++){
                if(laser == readLaser){
                    synchronized(readings){
                        for (int j = 0; j < photoCount * 2; j++)
                            readings[i][j] = readCounter();
                    }
                }
                laser *= 2;
            }
        }
    } catch(Exception e){
        e.printStackTrace();
    }
}

public int[][] getReadings(){
    synchronized(readings){
        int [][]result = new int[lasersCount][photoCount*2];
        for(int i=0;i<lasersCount;i++)
            for(int j=0;j<photoCount*2;j++)
                result[i][j]=readings[i][j];
    }
    return readings;
}

int readCounter() throws Exception{
    int result=0;
    for(int i=0;i<countersSize;i++){
        result += in.read() << (i*8);
    }
    return result;
}
}

```



### B-3. MATLAB code for coordinate calculation and 3D plotting

```
lr.stopReading();
close all;
LasersCount = 3;
PhotoCount = 4;
CounterSize = 4; % in bytes
CommPort = 'COM4';
BaudRate = 57600;
resolution = 10252; % without counting on both edges.
len = 39.92 % in inches

% calculate coordinates of end effector from coordinates of centroid of tetrahedron when
% the end-effector is parallel to the wing skin.
xadjust = 4.50 + 2.3465; % in inches
yadjust = -2.24 + 0.4522;
zadjust = -6.20;

xcline = [0 0 0 0 0 0 0 0];
ycline = [-26 -26 -26 26 26 26 26 -26];
zcline = [-36 0 0 0 0 -36 -36 -36];
xfill = [-2 2 2 -2];
yfill = [-2 -2 2 2];
zfill = [0 0 0 0];
C = [1 1 1 1];

javaaddpath('C:\Work\Matlab_code\commapi\comm.jar');
javaaddpath('C:\Work\Matlab_code\lasersreading\classes');
lr = lasersreading.LasersReader(CommPort, BaudRate, LasersCount, PhotoCount,
                                CounterSize);

lr.startReading()

h=figure;

while(1)

    Readings = double( lr.getReadings() );
    RawAngleReadings = Readings*2*pi/resolution;

    for i = 1:LasersCount

        sum = 0;
        SumWeight = 0;
        for j = 1:PhotoCount

            weight = Readings(i,j) - Readings(i,j + PhotoCount);
            SumWeight = SumWeight + weight;
            sum = sum + (weight* ( RawAngleReadings(i,j) + RawAngleReadings(i, j
                +PhotoCount) ) /2 );

        end

        Angles(i) = double(sum)/double(SumWeight);

    end

% change the cyclic order
LaserAngles(1) = Angles(2);
LaserAngles(2) = Angles(3);
LaserAngles(3) = Angles(1);
```

```

[x, y, z] = FindCoor(LaserAngles, len);
xac = x + xadjust;
yac = y + yadjust;
zac = z + zadjust;

figure(h);
% hold on
plot3(x, y, z, '^', 'MarkerSize', 5, 'LineWidth', 10);
axis ([-48 48 -26 26 -36 0])
title( strcat( ' X = ', num2str(xac * 25.4), ' mm', ' Y = ', num2str(yac *
25.4), ' mm', ' Z = ', num2str(zac * 25.4), ' mm' ), 'FontSize', 20
);

grid on;
hold on;
xlabel('x', 'FontSize', 20 );
ylabel('y', 'FontSize', 20 );
zlabel('z', 'FontSize', 20 );

line([-48 -48 -48 48 48 48 48 -48 48 48 48 -48 -48 -48],[-26 -26 -26 -26 -26
-26 -26 -26 -26 26 26 26 26 -26], [0 -36 -36 -36 -36 0 0 0 0 0 0 0
0 0 ], 'Color', 'r', 'LineWidth', 6)
line([-48 -48 -48 48 48 48 48],[-26 26 26 26 26 -26],[-36 -36 -36 -36 -36 -36],
'Color', 'r', 'LineWidth', 6, 'LineStyle', '-.')
line([-48 -48],[26 26],[0 -36],'Color', 'r', 'LineWidth', 6, 'LineStyle', '-
.')
line([48 48],[26 26],[0 -36],'Color', 'r', 'LineWidth', 6)
line([48 48],[26 -26],[-36 -36],'Color', 'r', 'LineWidth', 6)
line(xcline, ycline, zcline, 'Color', 'g', 'LineWidth', 3, 'LineStyle', ':');
fill3(xfill,yfill,zfill,C);
view(25,25);
hold off;

fid = fopen('C:\Work\Matlab_code\feedback\readings.dat', 'w');
fprintf(fid, '%6.1f %6.1f %6.1f', [xac,yac,zac]*25.4);
fclose(fid);

```

end

```
function [x, y, z] = FindCoor(angles, len)
```

```

if (angles(2) >= 3*pi/2)
    theta1 = angles(1);
    theta2 = angles(2) - (3*pi/2);
    theta3 = pi - angles(3);
    h = -len/(cot(theta1) + cot(theta3));
    x = h;
    y = (len/2) + h*cot(theta3);
    z = -h*cot(theta2);

```

```
else
```

```

    theta1 = 2*pi - angles(1);
    theta2 = (3*pi/2) - angles(2);
    theta3 = angles(3) - pi;

```

```

    h = -len/(cot(theta1) + cot(theta3));
    x = -h;
    y = (len/2) + h*cot(theta3);
    z = -h*cot(theta2);

```

```
end
```

```
end
```

# Bibliography

[1] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques", *Journal of Robotic Systems*, 14(4):pp.231--249, 1997.

[2] Andy Harter, Andy Hopper, Pete Steggles, Any Ward and Paul Webster, "The anatomy of context-aware application", proceedings of the 5<sup>th</sup> annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1999), pp. 59-68, 1999.

[3] Peter H. Dana, "Global Positioning System Overview", website, 2000.  
<http://www.colorado.edu/geography/gcraft/notes/gps/gps.html>.

[4] J. Hightower and G. Borriello, "*Location Systems for Ubiquitous Computing*", *Computer*, vol. 34, no. 8, Aug. 2001, pp. 57-66.

[5] E. Kruse, F. M. Wahl, "*Camera-Based Monitoring System for Mobile Robot Guidance*", *Proceedings of the IEEE/RSJ*, 1998, pp. 1248-1253.

[6] W. D. Rencken. "Concurrent Localisation and Map Building for Mobile Robots Using Ultrasonic Sensors" In *IEEE Int. Workshop on Intelligent Robots and Systems (IROS '93)*, 1993.

[7] R. Want et al., "The Active Badge Location System," *ACM Trans. Information Systems*, Jan. 1992, pp. 91-102.

[8] A. Harter et al., "The Anatomy of a Context-Aware Application," *Proc. 5th Ann. Int'l Conf. Mobile Computing and Networking (Mobicom 99)*, ACM Press, New York, 1999, pp. 59-68.

- [9] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," *Proc. 6th Ann. Int'l Conf. Mobile Computing and Networking (Mobicom 00)*, ACM Press, New York, 2000, pp. 32-43.
- [10] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," *Proc. IEEE Infocom 2000*, IEEE CS Press, Los Alamitos, California, 2000, pp. 775-784.
- [11] F. Raab et al., "Magnetic Position and Orientation Tracking System," *IEEE Trans. Aerospace and Electronic Systems*, Sept. 1979, pp. 709-717.
- [12] S.R. Bible, M. Zyda, and D. Brutzman, "Using Spread-Spectrum Ranging Techniques for Position Tracking in a Virtual Environment," *Second IEEE Workshop Networked Realities*, <http://www.npsnet.org/~zyda/pubs/NR95-Paper-Bible.pdf>.
- [13] E. Menegatti, A. Pretto, A. Scarpa, E. Pagello "Omnidirectional vision scan matching for robot localization in dynamic environments", *IEEE Transactions on Robotics*, Vol: 22, Iss: 3 ISSN: 1552-3098, June 2006, pages 523- 535
- [14] E. Menegatti, G. Gatto, E. Pagello, T. Minato, H. Ishiguro "Distributed Vision System for robot localization in indoor environments", *Proc. of the 2nd European Conference on Mobile Robots ECMR'05 September 2005 Ancona - Italy* pp. 194-199.
- [15] E. Kruse and F. M. Wahl," Camera-Based Monitoring System for Mobile Robot Guidance", *Proc. of IEEE, RSJ, Intl. Conference on Intelligent Robots and Systems*, Canada, Oct 1998, pp. 1248-1253.
- [16] J. Krumm et al., "Multi-Camera Multi-Person Tracking for Easy Living," *Proc. 3<sup>rd</sup> IEEE Int'l Workshop Visual Surveillance*, IEEE Press, Piscataway, N.J., 2000, pp. 3-10.

[17] T. Darrell et al., "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," Conf. Computer Vision and Pattern Recognition, IEEE CS Press, Los Alamitos, Calif., 1998, pp. 601-608.

[18] R.J. Orr and G.D. Abowd, "The Smart Floor: A Mechanism for Natural User Identification and Tracking," Proc. 2000 Conf. Human Factors in Computing Systems (CHI 2000), ACM Press, New York, 2000.

[19] D. Fox et al., "A Probabilistic Approach to Collaborative Multi-Robot Localization," Autonomous Robots, June 2000, pp. 325-244.

[20] Metris company website.

[http://www.metris.com/large\\_volume\\_tracking\\_positioning/basics\\_of\\_igps/](http://www.metris.com/large_volume_tracking_positioning/basics_of_igps/)

[21] Frank Y. S. Lin and P. L. Chiu "A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks", IEEE Communication Letters, Vol. 9, No.1, Jan 2005, pp. 43-45.

[22] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks", Proc. IEEE Wireless Communications and Networking Conference, pp. 1609-1614, 2003.

[23] B. Roy and HH. Asada, "Design of a Reconfigurable Robot Arm for Assembly Operations inside an Aircraft Wing-Box", Proc. of IEEE conference on Robotics and Automation, Spain 2005, pp. 590 – 595.

[24] Optech company website. <http://www.optech.ca/prodilris.htm>