

ARRIBOT- AN ENABLER FOR STUDYING MOBILE WIRELESS SENSOR  
NETWORKS

By

JAYMALA S. GHADIGAONKAR

Presented to the Faculty of the Graduate School of  
The University of Texas at Arlington in Partial Fulfillment  
Of the Requirements for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2006

Copyright © Jaymala S Ghadigaonkar 2006

All Rights Reserved

## ACKNOWLEDGEMENTS

I would like to acknowledge the following people who have helped me make this thesis possible.

First of all, I would like thank my supervising professor, Dr. Dan O. Popa for his valuable guidance and constant motivation. Without his deep involvement in my thesis work, this thesis would not be completed. I would like to take this opportunity to thank my supervising committee members Dr. Harry Stephanou and Dr Frank Lewis for gracefully agreeing to contribute their valuable time for the defense.

I am highly great full to my parents Mr. Suresh V. Ghadigaonkar and Mrs. Vijaya S. Ghadigaonkar for motivating me to achieve my goals and being there for me at time of downfalls.

Last but not the least, I would like to thank all the ARRI-DIAL members for making me feel comfortable and sharing their knowledge during my work on this thesis.

November 21, 2006

## ABSTRACT

### ARRI BOT: AN ENABLER FOR A MOBILE WIRELESS SENSOR NETWORK

Publication No. \_\_\_\_\_

Jaymala S. Ghadigaonkar, M.S.

The University of Texas at Arlington, 2006

Supervising Professor: Dr. Dan Popa

The objective of this thesis is to structure a mobile wireless sensor network using low cost, custom made robotic platforms. Over the years, several projects have proved that the performance of a static wireless sensor network (WSN) is greatly enhanced by using mobile WSN for applications like adaptive exploration or in search and rescue missions where the network topology changes with changes in the environment.

A low cost, mobile wireless sensor node known as the ARRI BOT has been designed to contribute along with other robotic platforms in structuring an ingenious mobile wireless sensor network (MSWN). The ARRI BOT has the ability to navigate

and localize and can extend its lifetime in a network by harvesting solar energy and vibration energy required for its operations. The robot has been tested to validate several robotic algorithms such as Adaptive Sampling, Localization using static beacons and navigation using obstacle avoidance.

This thesis has contributed in designing a second generation of ARRI BOT which incorporates Crossbow's crickets for wireless communication and localization, ultrasonic ranging module for obstacle avoidance, color sensor for validating sampling algorithms, servos, encoders for dead reckoning and a solar panel for charging the on-board batteries. Javelin stamp microcontroller controls the intelligence of the robot and the mentioned modules are interfaced to it. The second generation ARRI BOT has been also tested for fundamental localization technique using Crossbow's crickets.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
ABSTRACT .....	iv
LIST OF ILLUSTRATIONS.....	viii
LIST OF TABLES.....	xii
Chapter	
1. INTRODUCTION .....	1
1.1. Motivation for deploying wireless sensor network with robots .....	2
1.2. Challenges involved in designing a wireless sensor node .....	3
1.3. Introduction to ARRI BOT .....	6
1.4. Contribution.....	8
1.5. Thesis organization.....	9
2. LITERATURE REVIEW .....	10
2.1. Related work.....	10
2.2. Wireless Sensor Network (WSN).....	18
2.3. Localization .....	20
2.4. Adaptive Sampling .....	24
2.5. Evolution of ARRI BOT.....	26
3. DESIGN ARCHITECTURE OF ARRI BOT.....	28
3.1. Electrical design of ARRI BOT .....	28

3.2.	Structural design of ARRI BOT .....	58
3.3.	Software control for ARRI BOT .....	60
4.	TOOLS AND ALGORITHMS FOR DEPLOYMENT.....	63
4.1	Wireless Communication on ARRI BOT .....	63
4.1.1.	Challenges in wireless communication on ARRI BOT .....	68
4.2.	Localizing ARRI BOT using dead reckoning .....	71
4.2.1.	Localization of ARRI BOT using active beacons .....	73
4.2.2.	Challenges involved in localizing ARRI BOT .....	74
4.3.	Sensor Fusion for localization .....	74
4.4.	Adaptive Sampling using ARRI BOT .....	77
5.	EXPERIMENTAL RESULTS .....	88
5.1.	ARRI BOT in Localization.....	88
5.2.	ARRI BOT in energy harvesting.....	97
5.3.	ARRI BOT in Adaptive sampling .....	98
6.	CONCLUSION.....	104
6.1.	Thesis contribution .....	104
6.2.	Future work .....	105
Appendix		
A.	ELECTRICAL DESIGN OF ARRI BOT.....	107
B.	STRUCTURAL DESIGN OF ARRI BOT .....	110
REFERENCES	.....	126
BIOGRAPHICAL INFORMATION.....		133

## LIST OF ILLUSTRATIONS

Figure	Page
1.1 The ARRI BOT.....	7
2.1 PolyBot in a snake, loop and spider form.....	11
2.2 University of Southern California's Robomote.....	12
2.3 MOBILE Robot's Amigobot.....	12
2.4 YUPPY & Pebble.....	13
2.5 Acroname's Garcia.....	14
2.6 Carnegie Mellon University's Tag Bots use Intel boards.....	15
2.7 University of North Carolina-Charlotte's Explore Bot.....	15
2.8 University of Oulu's Minirobot.....	16
2.9 K-Team's Khepera – III.....	17
2.10 Autonomous Mini Robot "Alice 2002".....	18
2.11 Localization using signal from four satellites.....	23
2.12 ARRIBOT-I with Parallax Transceiver.....	26
2.13 ARRIBOT-II with MICA2 transceiver and ranging module.....	26
2.14 ARRIBOT-II with Cricket as transceiver.....	27
2.15 ARRIBOT-II prototype with a customized PCB.....	27
2.16 ARRIBOT-II prototypes, with a new chassis and a solar panel.....	27
3.1 Block diagram of ARRI BOT.....	28



3.2	Javelin Stamp Microcontroller .....	29
3.3	Block diagram of motor control .....	31
3.4	Picture of Futaba’s S148 servo and internal circuitry of the servo.....	32
3.5	Picture of the first generation optical encoders and the second generation optical encoders. ....	32
3.6	TCS230 Color Sensor Evaluation Kit.....	34
3.7	Block diagram of color sensor interface with CPU .....	34
3.8	Ultrasonic Transducer (SRF04).....	35
3.9	Beam Pattern of the Ultrasonic Transducer.....	36
3.10	3-D representation of the beam pattern .....	37
3.11	Block diagram of ultrasonic interface to CPU.....	38
3.12	Timing Diagram of the Ultrasonic Transducer (SRF04).....	39
3.13	Spectrum of Sunlight .....	41
3.14	Solar Panel of ARRIBOT .....	42
3.15	Circuit for solar energy management .....	43
3.16	Block Diagram of the step- down converter (LM2574).....	44
3.17	Characteristic Curves for the Inductor Selection.....	46
3.18	Battery and Javelin Stamp interface .....	47
3.19	Piezoelectric panel.....	48
3.20	Parallax card, MICA2, Cricket.....	49
3.21	MICA2’s 51 pin connector .....	50
3.22	MICA2 and ARRIBOT interface.....	50
3.23	Cricket Sensor Node.....	51

3.24	RS232 Communication Interface .....	55
3.25	Power Distribution.....	56
3.26	Schematic diagram of ARRI BOT.....	57
3.27	PCB layout of ARRI BOT .....	57
3.28	Structural Model of ARRIBOT .....	58
3.29	Optimized mounting of the ultrasonic modules.....	59
3.30	Snap shot of Javelin Stamp IDE .....	60
3.31	Snap shot of Labview based user interface.....	60
3.32	Flowchart of software control.....	61
4.1	Block diagram of wireless communication between base station and ARRI BOT .....	63
4.2	TinyOS packet formation procedure .....	67
4.3	Multi-hopping with six robots .....	69
4.4	Tricycle robot configuration.....	72
4.5	Localization of ARRI BOT using triangulation .....	73
4.6	Block diagram of sensor fusion .....	75
4.7	Linear parametric field for validating Adaptive sampling .....	77
5.1	Experimental setup for testing localization accuracy .....	89
5.2	Actual verses and measured distance obtained from the crickets.....	89
5.3	Experimental Setup for testing localization using triangulation.....	91
5.4	Plot of actual and measured position with correction factor .....	92
5.5	Experimental Setup with three robots.....	93
5.6	x y position recorded from three robots at different time instant .....	94

5.7	Experimental Setup for localization demo .....	95
5.8	The VI characteristic of the solar panel.....	97
5.9	Experimental setup for testing adaptive sampling.....	99
5.10	Simulation results from raster scanning .....	100
5.11	Simulation results from adaptive sampling .....	100
5.12	Simulation results from greedy adaptive sampling .....	101
5.13	Experimental results from raster scanning .....	102
5.14	Experimental results from adaptive sampling (AS).....	102
5.15	Experimental results from greedy adaptive sampling (GAS).....	103

## LIST OF TABLES

Figure	Page
3.1 Fixed point representation of a floating point number .....	31
3.2 Technical specification of the servos.....	32
3.3 Properties of Ultrasonic Transducer .....	36
3.4 Efficiency achieved with different types of solar cells.....	41
3.5 Specification of solar panel on the ARRI BOT .....	42
3.6 RS-232 Specifications .....	53
3.7 RS 232 DB25 pin description.....	54
3.8 RS 232 DB9 pin description.....	54
3.9 Power Consumption budget of the modules on ARRI BOT .....	56
3.10 Different mounting combination of ultrasonic modules.....	59
3.11 List of software command for ARRI BOT .....	62
4.1 A typical TinyOS packet .....	64
4.2 A typical Payload data Packet .....	65
5.1 Comparison of the actual distance and distance from the crickets.....	90
5.2 Time taken to charge by the batteries .....	98
5.3 Simulation comparison of different sampling methods.....	101
5.4 Experimental comparison of sampling algorithms .....	103

## CHAPTER 1

### INTRODUCTION

Wireless sensor networks (WSN) are an active research area due to their unique combination of being distributed and wireless and their ability to satisfy data acquisition and processing constraints. WSN development was originally motivated by military applications for battle field surveillance. However their applications are varied and unlimited. They replace the conventional wired sensor networks which are expensive to maintain and have limited applications. Typical applications of WSN are in monitoring and surveillance [3] for instance, in environmental monitoring, habitat monitoring, acoustic detection, seismic detection, military surveillance, inventory tracking, medical monitoring, smart spaces, process monitoring etc.

However the performance of a static wireless sensor networks declines in applications like adaptive exploration and search and rescue missions, where the network topology changes dynamically. Also they are generally supplied with limited power which in turn offers a limited operating time. However, by adding mobility to a static WSN these drawbacks can be addressed by exploiting the re-configurability, redundancy, multi-hopping characteristics of a mobile wireless sensor network.

## 1.1 Motivation for deploying wireless sensor network with robots

### *Spatio-temporal irregularities*

Research in static wireless sensor network [1] has shown that there exist spatio-temporal irregularities in sampling with static wireless sensor nodes. Spatial irregularities occur either due to uncertainty in the phenomena of interest and the variation in the distribution of the deployed sensor nodes or due to terrain irregularities and practicalities the sensor nodes are generally deployed in places where there is access to power and communication. Temporal irregularities on the other hand are caused due to lack of synchronization between the sensor nodes in the network.

A densely deployed sensor node to obtain large sampling volumes is an expensive approach and increases the overhead on power consumption. However with addition of controlled mobility to a wireless sensor node for spatial adaptation helps in reducing the spatial irregularities. Temporal irregularities however needs more complex expertise and is still under active research.

### *Adaptable Deployment*

Static wireless sensor nodes lack the ability to change their deployment topology based on dynamic changes in the environment. Thus they fail in applications like the search and rescue mission, military surveillance etc. or in case of variation in communication range and lack of energy. However adding mobility to these nodes solves these problems and also facilitates execution of self-deploy and self-repair [2] algorithms in a dynamic environment.

### *Energy harvesting*

In order to have a self sustaining sensor network it is essential to recharge the nodes from time to time. Several methods have been suggested to transport power from a point where there is profuse supply to point where there is scarcity of power. Another alternative is to harvest natural hand-on energy (solar or vibration) to charge the batteries on the sensor node for achieving a continuous operation [7].

### *Event based environment monitoring*

One of the classic applications of a static wireless sensor network is to detect occurrences of an event during environment monitoring. However the distribution of the static nodes is fixed while the distribution of an event is dynamic. In order to ground these events using static WSN requires massive deployment of sensor nodes which is not a cost effective solution. However by making the nodes mobile, dynamically distributed events can be captured in a cost and energy effective way [47].

## 1.2 Challenges involved in designing a wireless sensor node

Although a mobile wireless sensor network (MWSN) addresses the drawbacks of a static WSN, structuring a reliable, self-sustaining mobile wireless sensor network involves several challenges.

### *Communication range*

The most important feature of any mobile wireless sensor network is the area that is covered. Every wireless communicating module comes with a limited range of coverage and this limited range is further affected by obstacles present in the environment such as walls in an indoor infrastructure. To get around this limitation,

techniques like multi-hopping have been suggested, that help in extending the networks operating range.

### *Localization*

Localizing the sensor nodes is a key element for an effective mobile wireless sensor network. For applications like the search and rescue mission, it is imperative to know the location of the nodes in order to direct the people to the nearest safe exits. Localization has been an active research topic in the field of distributed robotics. Software based approaches like the probabilistically based localization [5] based on Bayes's rule and EKF based adaptive sampling have undergone immense research. However, such approaches involve lot of computations. Another approach is to use active beacons to localize the listener, but they are generally expensive to maintain and are not Omni directional. A hybrid approach can be structured to offset the weaknesses of each other.

### *Power management*

In general, mobile sensor nodes are equipped with a portable battery packs that last for couple of hours based on the application. It is been observed that the performance of the wireless sensor node deteriorates with drop in battery voltage. Several energy efficient algorithms [4] have been developed to support ongoing operation of the sensor network. Another approach is to use naturally available energy, such as, solar, vibration, wind, and motion etc to charge the batteries on the node.



### *Extensive programming*

Extensive programming is involved to manage a mobile wireless sensor network with multiple nodes. Programming each node itself is time consuming. Also with several incoming and outgoing packets (for e.g. data packet or command packet or acknowledgement packets) the base station has to keep a track of each of them and process it based on the application. Also a flexible and modular based software programming has to be maintained so that it is easily expandable and changeable based on applications.

### *Sensor calibration*

Generally, the sensors on mobile nodes are cheap but provide output with some amount of discrepancy. Therefore, there is a need to calibrate them frequently or meticulously design them in software so that the errors can be counterbalanced.

The servos on the mobile nodes also need calibration with respect to their speed. They have to be calibrated such that at a given time, equal distance is traveled by both the wheels, to minimize errors during dead reckoning.

### *Cost*

Mobile wireless sensor nodes are deployed in large numbers and work under detrimental environments. Commercially available wheeled robots, for instance, Acroname's Garcia are comparatively expensive in the range of \$2500 - \$5000. The cost further increases with more advanced robots. An ideal mobile wireless sensor network comprises of hundreds of nodes, for this reason the cost in building them

should be kept low so that there can be a massive deployment and can be expendable if necessary.

### *Size*

The size to some extent controls the application of a mobile node. A smaller node helps the user to develop a complex environment in a small area and aid mobility. Also a large amount of energy is saved which is otherwise utilized in driving a large and heavy robot.

### *Mobility*

The application area of a robot is depended on the mobility mechanism it uses. Surface mobility can be achieved through wheeled, tracked or legged robots. A lot of research is currently being done in exploring new breeds of mobility such as crawling, burrowing, hopping, swimming and even flying robots. These diverse mobility help in traversing in a rough terrain and also explore different environmental mediums such as air and water.

## 1.3 Introduction to ARRI BOT

ARRI BOT is a low cost mobile wireless sensor node that was developed in the Distributed and Intelligence and Autonomy Lab (DIAL) at the Automation and Robotic and Research Institute (ARRI). It was envisioned with an objective to contribute along with other robotic platform in structuring an indoor Mobile Wireless Sensor Network, as well as to add essential functionalities like localization, navigation and power management while keeping the cost low as USD 600.

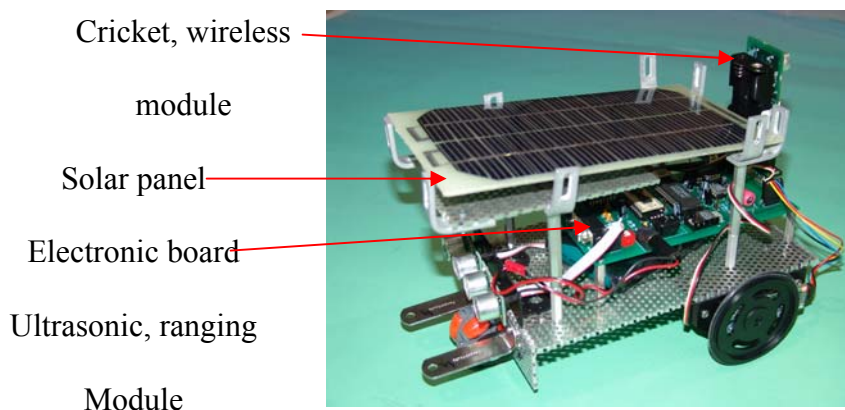


Figure 1.1 The ARRI BOT

#### *ARRI BOT Features*

- ❖ Compact electrical design achieved by the use of multilayer PCB layout and on-board micro-controller circuits
- ❖ High Precision maneuverability of resolution less than 2mm achieved by continuous servo motors and optical encoders
- ❖ Precision control over speed and direction achieved using Pulse Width Modulation (PWM) controlled servo motors
- ❖ Onboard sensor package comprising of an inbuilt color sensor and cricket sensor board
- ❖ Wireless communication protocol incorporated using cricket as transceiver
- ❖ Collision avoidance module implemented using ultra sonic range finders

- ❖ Continuous power level monitoring and range updating
- ❖ Alternate source of energy such as solar cells for charging the on-board battery.
- ❖ Light weight, low cost and compact design ideal for safe, large scale deployment

#### 1.4 Contribution

In order to address the challenges mentioned in the previous section and in order to test out deployment algorithm in our lab, we designed and built a low cost, mobile wireless sensor node, the ARRI BOT.

My contribution to this research was as follows

- ❖ A low cost mobile wireless sensor node called '*ARRIBOT*' has been designed, assembled and tested.
- ❖ Advanced wireless protocols have been programmed to establish communication between the base station and the mobile sensor nodes using Crossbow®'s Crickets.
- ❖ Onboard sensor package has been upgraded to multiple sensors comprising of light, temperature, vibration sensors etc. along with the prevailing color sensor
- ❖ Dynamic collision avoidance system has been implemented on the robot using ultrasonic range finders.
- ❖ Advanced localization schemes have been implemented for multiple mobile robot deployment scenarios.

❖ Power optimization schemes such as on-board power monitoring and alternative energy harvesting through solar cells are also implemented into the mobile robot.

❖ PC based user interface program has been written in Java® and Matlab® in order to navigate and localize the robot and also to process sensor data.

### 1.5 Thesis organization

The thesis is organized as follows:

Chapter 2 introduces some contemporary robotic platforms, similar to ARRI BOT, which are either designed at different universities or commercialized for research purposes. It also provides a brief history of ARRI BOT and its evolution over time.

Chapter 3 discusses the electrical and structural designing of ARRI BOT in great detail.

Chapter 4 presents deployment algorithms such as the fundamental wireless communication, localization and adaptive sampling, which have been tested using ARRI BOTs.

Chapter 5 presents the results and observation obtained from experimenting localization and adaptive sampling algorithms and also the performance of energy harvesting on ARRI BOTs.

Chapter 6 presents the conclusion made from this research and also provides some suggestions for future work.

## CHAPTER 2

### LITERATURE REVIEW

Mobile robotics is been an active research field for the last two decades. Today's robots can demonstrate unusual maneuverability like leaping, scrambling, rolling, climbing or even flying. This particular chapter presents a brief introduction to some robotic platforms in the category of the classic wheeled robots. It also gives a brief introduction of ARRI BOT and its evolution over time.

#### 2.1 Related Work

Robots can be broadly classified as mobile robots, manipulator robots and self reconfigurable robots [6].

- *Mobile robots* – They are automatic machines that have the ability to move in a given environment. A big area of research today is in mobile manipulators such as the humanoid robots. The most common class or mobile robots are the wheeled robots. The second class is the legged robots while the UAV (Unmanned Aerial Vehicles) and underwater vehicles form another interesting but small class of robots.

- *Manipulator robots* - They are mainly industrial robots and are programmed to faithfully carry out specific actions repeatedly with high degree of accuracy. The SCARA [53] (Selective Compliant Articulated/Assembly Robot Arm) and the PUMA 500 [54] are examples of manipulator robots.

▪ *Mobile manipulators* – Mobile manipulators are a combination of a mobile platform and an on-board manipulator. Humanoid robot such as Honda’s ASIMO, Robovie-M [66], Androids and Gynoids are some examples of mobile manipulators [65].

▪ *Self reconfigurable robots* – It refers to the use of numerous semi-independent modules that can combine themselves in different ways to perform different functions. PolyBot [46] is an example of self reconfigurable robots.



Figure 2.1 PolyBot in a snake, loop and spider form [46]

Wheeled robots are the classic and popular example of mobile robots. Robotic theories like navigation, path planning, trajectory following, localization, sampling and many more have been studied using wheeled robot. Mentioned below are some examples of contemporary wheeled robots which are either commercialized or designed for research purposes.

### *Robomote*

The second generation Robomote is a mobile wireless sensor node developed at the University of Southern California [7]. The Robomote works in cooperation with the UC Berkeley's mote to achieve wireless communication. Robomote-I had a modular interfaces for the radio component, while Robomote II has interfaces for actuation that it can use to drive itself around.

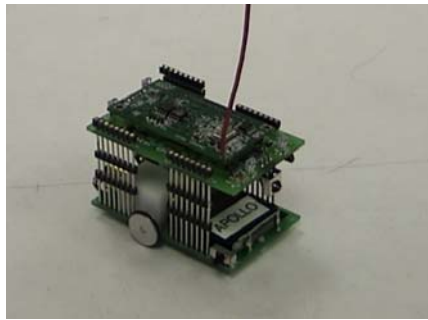


Figure 2.2 University of Southern California's Robomote [7]

### *AMIGOBOT*

Mobile Robot's AMIGOBOT [8] is a commercially available robot designed to serve multi-robot and classroom applications. It is used by the DARPA teams for large-scale robot collaboration research.



Figure 2.3 Mobile robot's Amigobot [8]



It is house an on-board Renesas SH7144-based microcontroller. Amigobot is equipped with two 2 drive wheels and motors, rear caster, speaker, long-life, rechargeable battery and rugged molded plastic shell. It can be programmed using C/C++ under Linux or WIN2000/XP. It is equipped with six forward and two rear ultrasonic range finders.

*YUPPY & Pebble*

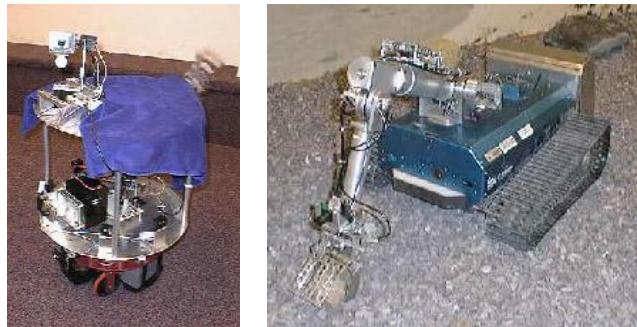


Figure 2.4 YUPPY & Pebble [9]

Yuppy and Pebble are employed at the MIT's Artificial Intelligence Lab [9]. Yuppy is programmed to respond emotionally to situations experienced in the world and to human interacting with it. Yuppy has four drives relating to battery charge, temperature, energy, and interest levels.

Pebble III was designed by the IS Robotics, Inc. It is a vision-based mobile robot that uses a single camera for obstacle avoidance in rough, unstructured environments built with an objective to explore the Martian surface by using visual cues to avoid rocks and craters too large to traverse.

### *Garcia Robot*

Garcia robot is a commercially available robot designed by Acroname Inc. [10].



Figure 2.5 Acroname's Garcia [10]

It is the most sophisticated and high quality robot developed to help the user to concentrate primarily on high-level programming. It is equipped with two onboard processors operating at 40MHz, a BrainStem Moto 1.0 processor to handle the motion control and several sensor inputs and a BrainStem GP 2.0 processor to manage a serial interface, IR communication capability, and additional I/O.

It houses two highest quality Maxon motors with built-in 16 counts per rotation quadrature encoders and 19:1 planetary gearboxes. An extra 3:1 gear reduction stage between the motor and wheel supports the weight of the robot. Odometry is achieved with 3648 encoder pulses per wheel rotation. It has six IR range finders to provide distance measurements in a range of 4 to 18 inches.

### *TagBot*

TagBots are built at Carnegie Mellon University [11]. It is a sophisticated robot incorporating Stayton-CMUcam communication.

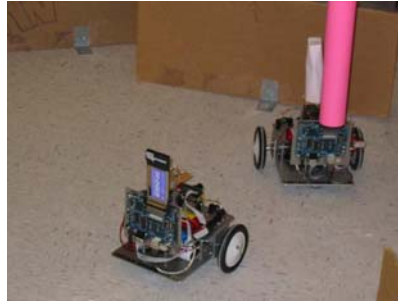


Figure 2.6 Carnegie Mellon University's TagBots use Intel boards [11]

As shown in the picture above, two robots play tag, chasing each-other around by looking at the bright red/orange paper cylinder on each robot. The one that is not "it" just wanders aimlessly, using range finders to avoid walls. The one that is "it" spins in a circle until it sees the bright pink hat on the other. It uses a CMUCam for vision. When it sees its prey, it follows it. Once its range finders tell it that there is something in front of it and it can see the other robot, it sends a message over the wireless network to the other robot that causes them to switch roles. The one that was tagged blinks and waits several seconds to give the other a chance to get away.

### *Explorebot*

Explorebot was designed at The University of North Carolina-Charlotte [12], as a dynamic outreach for an NSF-funded Girl Scouts project.

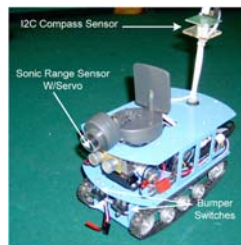


Figure 2.7 University of North Carolina-Charlotte's Explorebot [12]

The objective was to build a low cost, flexible autonomous mobile robot to be used for a wide range of indoor experimentation on multi-hop ad hoc and sensor networking. It employs a powerful Rabbit Semiconductor R3000 on-board microprocessor for accurate robotic movement control, MICA for wireless communication, an on-board wireless camera and built-in compass and ranging modules for navigation. The cost involved in assembling the robot is approximately \$1,100.

### *Minirobot*

Minirobot [15] was designed at the University of Oulu, Oulu, Finland with the objective to have multiple small size robots for saving valuable work space and making the maintenance of the robots easier.



Figure 2.8 – University of Oulu's Minirobot [15]

The Minirobot is equipped with an 8-bit low power 8MHz MCU (ATmega32). The radio module on Minirobot is based on nRF905 multi-band transceiver designed for 433/868/915MHz, with a 512kB on-board Flash memory for temporarily storing uploaded application code. The size is approximately 80mm in diameter and 34mm in height.

### *Khepera*

The K-Team's Khepera III architecture provides exceptional modularity [44]. As any K-Team robot, the Khepera III is using an extension bus system for nearly unlimited configurations.



Figure 2.9 – K-Team's Khepera-III [44]

The robot base includes an array of 9 Infrared Sensors for obstacle detection as well as 5 Ultrasonic Sensors for long range object detection. It also proposes an optional front pair of ground Infrared Sensors for line following and table edge detection. The robot motor blocks are Swiss made quality mechanical parts, using very high quality DC motors for efficiency and accuracy. The swappable battery pack system provides a unique solution for almost continuous experiments, as an empty battery can be replaced in a couple of seconds. The size is approximately 70mm in diameter and 30mm in height. The cost of Khepera-III is around USD 1350.

### *Alice*

Swiss Federal Institute of Technology Lausanne's (EPFL), "Alice 2002" was envisioned with an objective to design an inexpensive, small intelligent mobile robot which would aid in studying collective behavior with large quantity of robots [45].

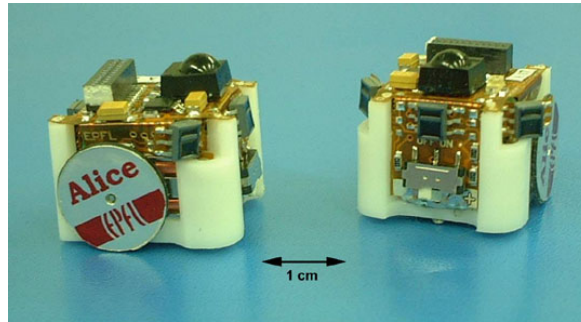


Figure 2.10 Autonomous Mini Robots "Alice 2002" [45]

It is equipped with two SWATCH motors with wheels and tires, a PIC16LF877 CPU, four active IR proximity sensors and a transceiver for remote control.

## 2.2 Wireless Sensor Network (WSN)

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations [16]. Wireless communication and mobile data sharing methods have evolved tremendously over the past few years. They have potential application for military or commercial use especially in places where no communication infrastructure exists. Some typical applications of WSN are as follows;

- Environmental monitoring.
- Habitat monitoring.
- Military surveillance.
- Traffic monitoring.
- Nuclear reactor controlling.

A wireless communication network consists of several nodes communicating with a remote base station and also among them. Some of the basic topologies for structuring a network are fully connected mesh, star, ring, tree and bus networks. With multiple nodes transmitting and receiving packets over the network, there is potential possibility of collision. Several protocols have been developed to avoid collision, such as, the ALOHA, Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA) [63]. Some of the other challenges involved in structuring a WSN are detecting relevant events, monitoring and sampling the data of interest, assessing and evaluating the sensor information, formulating a meaningful user interface, and performing decision-making and alarm functions, if required [63].

Protocols are also developed to support ad hoc wireless communication which include the Bluetooth, DECT (Digital Enhanced Cordless Telecommunications), DSRC (Dedicated Short Range Communications), HIPERLAN, HIPERMAN, IEEE 802.11, IrDA, RFID (Radio Frequency Identification), Ultra-wideband (UWB from WiMedia Alliance), WiFi, WiMAX, xMax, ZigBee, 3G, 3GPP, Wireless USB. Some of the current wireless systems are discussed here [16].

### *IEEE 802.11*

The Institute of Electrical and Electronic Engineers (IEEE) 802 working group are responsible for the development of LAN standards. The IEEE 802.11 which denotes Wireless LAN (WLAN) was developed by IEEE's working group 11. WLAN provide flexibility over the conventional wired communication especially in establishing

distance communication. Crossbow's Crickets are based on the IEEE 802.11 standard and used on ARRI BOT to achieve wireless communication.

### *Bluetooth*

Bluetooth is an industrial standard for wireless personal area networks i.e. PAN [16]. It was designed to achieve low power communication over a short range. The common applications of Bluetooth are in establishing wireless communication between cell phones and hands free headset, wireless networking between PCs over a limited bandwidth and space, wireless connection between PDA and a PC etc. Bluetooth system operates in master- slave mode, where the master can communicate with up to seven slaves. A network with one master and seven slaves is called a Piconet. At a given time, the master establishes communication with one slave and switches to other slaves in a round robin fashion.

## 2.3 Localization

Localization is a process of determining or marking the location of the mobile robots in its environment. Localization has been a critical underlying factor for successful mobile robot navigation in a copious and dynamic environment and has been researched tremendously in the past few years [25]. Measurements used to localize the robots can be broadly categorized as

- Relative position measurements.
- Absolute position measurements.



### *Relative position measurements*

Relative position measurements are based on the speed, direction, time interval between the past and present measurement. This technique is widely used in localizing ships, aircrafts, automobiles, rail engines, construction site engines (e.g. in tunnels) and mobile robots. Odometry and inertial navigation are the example of relative measurements.

### *Localization using odometry measurements (Dead Reckoning)*

Dead reckoning (DR) is the process of estimating a global position of a vehicle by advancing a known position using course, speed, time and distance to be traveled.

Due to various uncertainties DR faces two different types of errors, namely systematic and non systematic.

Systematic errors are those caused due to problem inherent in the robot independent of environment effects. These could be due to slight inequality in the speed of the two motors driving the two wheels, difference in the diameter of the wheel or the effect of drop in battery voltage powering the motors [24]. However effects of these errors can be reduced by meticulously modeling and filtering them out.

Non systematic errors are those caused due to changes in the robots environment such as variation in floors roughness, bumps and unevenness or it could be caused due wheel slippage etc. These factors are unpredictable and can affect the kinematic relation between the distance covered and the amount of wheels rotation.

### *Localization using Inertial Navigation*

An inertial navigation system (INS) provides the position, velocity, orientation, and angular velocity of a vehicle by measuring the linear and angular accelerations applied to the system in an inertial reference frame [28]. Inertial guidance systems were originally developed for navigating rockets. An INS integrates the information gathered from a combination of gyroscopes and accelerometers in order to determine the current state of the system. A gyroscope detects small accelerations in orientation while the accelerometers measures small accelerations along the x or y axis of the robot.

Similar to odometry measurements INS is calculated by integrating information from the sensors. However they suffer extensive drift and are sensitive to bumpy grounds.

### *Absolute position measurements*

Absolute position measurements give the complete position information independent of the previous location estimate. Unlike the relative measurement method the location is not obtained by integrating a sequence of measurement and hence the error in measurement is not accumulated. Some of the methods for absolute position measurements are GPS, active beacons, or map based localization. With active beacons, the listener uses techniques like triangulation or trilateration to determine the absolute position. Triangulation uses distances and angles to three or more active landmarks while trilateration technique uses only distance information from the beacons.

### *Localization using Global Positioning System (GPS)*

The most popular used localization technique for outdoor navigation is the GPS. A GPS receiver uses a process called as trilateration that obtains the distance from at least four satellites to localize. There are at least 24 GPS satellites and some spares operated by the U.S Department of Defense [23].

Each GPS satellite transmits its location and the current time to the receiver in a synchronized fashion along with the other satellites to keep the retransmission of the signal at the same instant. This signal arrives at a GPS receiver at slightly different intervals because some satellites are further away than others. By determining the amount of time it takes for the signal to reach the receiver, the distance to a particular satellite can be estimated. The receiver can calculate its position in latitude, longitude and elevation by triangulating the distances obtained from at least four satellites.



Figure 2.11 Localization using signals from four satellites [22]

Strengths - GPS provide absolute position information with respect to latitude and longitudes all around the world.

Weakness – It is inaccurate for non military applications. Also it requires line of sight between the receiver and the satellites so it highly unsuitable for indoor or urban environments.

#### 2.4 Adaptive Sampling

Sampling is part of statistical practice concerned with the selection of individual observations intended to yield some knowledge about a concerned environment. However for dynamic and unknown environment, an adaptive sampling approach is employed where the sampling strategies are based on prior measurement or analysis. Adaptive sampling can be defined as a sampling method in which sampling regions are selected based on values of the variables of interest observed from previous samples [59]. The key goal of adaptive sampling is to exploit the subject's characteristics to obtain precise estimate of the next sampling stage. Some of the applications of adaptive sampling are, monitoring and mapping dynamic environments or carrying out planetary explorations like the NASA's Mars Exploration Rovers – Spirit and Opportunity or in search and rescue operations where the mobile node can transmit back the sampled data to a remote base station.

Mobile wireless sensor nodes are best suited for environment monitoring. The monitoring could include estimating the salinity of lake water, or air pollution over an industrial area etc. A Networked Infomechanical Systems (NIMS) [59] approach uses adaptive sampling to monitor spatiotemporally distributed events in a forest environment by mapping environmental variables of temperature, humidity, and solar illumination. NIMS enable active physical reconfiguration of a diverse spatiotemporal

distribution of sensor nodes in three dimensional environments. A combination of uncertainty in localization as well as in sensor measurements to achieve effective adaptive sampling has also been proposed [60] [61] using solar AUV (Autonomous Underwater Vehicle).

Several robot team tasks such as, foraging, ant colony behavior, robotic soccer, area searching, mine sweeping, etc. have been proposed for goal attainment and sensor network deployments using potential fields [62].

An extended Kalman filter based sampling algorithm to estimate the parameters of a linear parametric field is introduced in this thesis, to minimize the state uncertainty, represented by the covariance matrix.

A greedy approach is also introduced in the thesis for Adaptive sampling algorithm where the next sampling location is searched within the vicinity of the currently sampled location. The algorithm selects the next best grid to sample in order to minimize the error covariance. The advantage of greedy adaptive sampling is that less time is required to recreate the field and the robot covers less distance thereby saving energy. Greedy adaptive sampling lies between raster scanning and adaptive sampling in terms of number of samples and takes least time to estimate the field.

## 2.5 Evolution of ARRI BOT

The first generation of ARRI BOT (2004) integrated all the elemental modules such as the Javelin Stamp processor, servos and optical encoders for dead reckoning, a color sensor for validating different sampling method and a Parallax transceiver module to achieve fundamental wireless communication [47] [48].

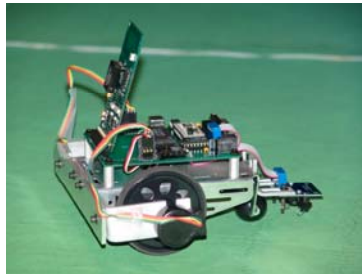


Figure 2.12 ARRIBOT-I with Parallax Transceiver

Subsequently the Parallax transceiver module was replaced with UC Berkeley's MICA2 radio module to achieve reliable communication and implement complex communication protocols such as multi-hopping. Also an ultrasonic ranging module was integrated to facilitate navigation in presence of obstacles (Jan 2006).

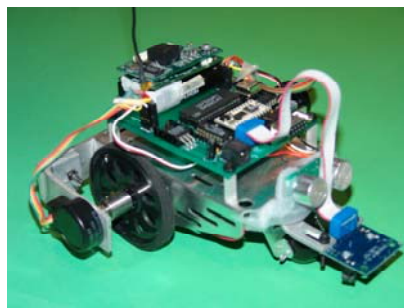


Figure 2.13 ARRIBOT-II with MICA2 transceiver and ranging module

To achieve localization, the MICA2 were then replaced with Crickets and an additional range finder was added to gain a broader coverage during navigation (May

2006). Crickets offered an excellent tradeoff between cost and functionality (combination of localization and communication in a single module).

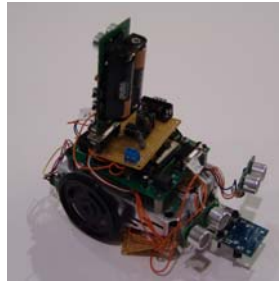


Figure 2.14 ARRIBOT-II with Cricket as transceiver

A new PCB was then designed to accommodate the additional interface with the ranging module, Cricket, Solar panel and also a set of new optical encoders (July 2006).

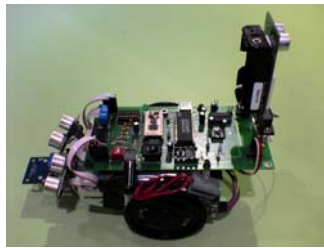


Figure 2.15 ARRIBOT-II prototype with a customized PCB

Finally, the old chassis was been replaced to house the new electronic board, solar panel, color sensor, ranging modules and an Omni directional front wheel (Sep 2006).



Figure 2.16 ARRIBOT-II prototypes, with a new chassis and a solar panel

## CHAPTER 3

### DESIGN ARCHITECTURE OF ARRI BOT

In this chapter, the hardware and software design of ARRI BOT is described in great detail. Each module on the BOT, such as, the solar panel, ultrasonic range finder, Crickets, chassis etc are explained with respect to their functionality and interface with the BOT.

#### 3.1 Electrical design of ARRI BOT

The second generation of ARRI BOT was envisioned with an objective to contribute along with other robotic platform in structuring an indoor Mobile Wireless Sensor Network, as well as add in essential functionalities like localization, navigation, power management etc while keeping the cost low.

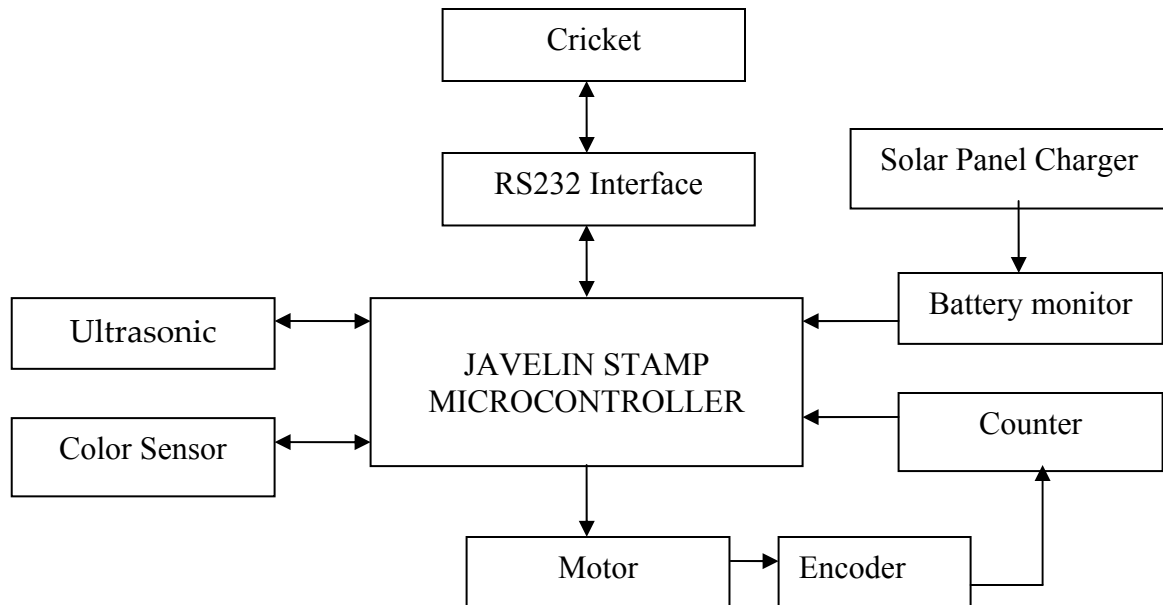


Figure 3.1 Block diagram of ARRI BOT



The electronic design of ARRI BOT can be outlined as independent module such as the ultrasonic based ranging module, Cricket based wireless communication module, color sensor, servos and counters etc, interfaced to an on-board CPU. Javelin stamp microcontroller forms the main processor and controls the modules through its I/O pins. Since Javelin stamp is equipped with only sixteen I/O pins, it is highly impossible to dedicate these pin to each module. Therefore, Javelin stamp selects between the ranging, color and motor- counter modules through a 3:8 multiplexer. The Cricket wireless module is selected at all times using Javelin's UART, which runs in the background to receive packets uninterruptedly.

#### *Processing Unit*

The intelligence of ARRI BOT to carry out various maneuvers is regulated by on- board processing unit called Javelin Stamp microcontroller. The drive to use Javelin Stamp is to have a cost effective solution towards achieving robotic maneuvers by means of easy, structured and object oriented Java based programming.

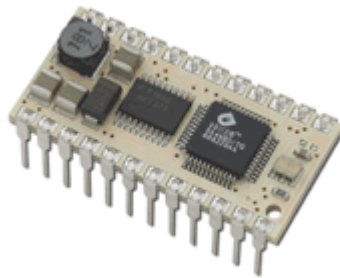


Figure 3.2 Javelin Stamp microcontroller

### *Features of the CPU*

- Javelin Stamp is a 24pin DIP module incorporating a Ubicon SX48AC microcontroller, 16 digital input/output pins and uses Java as its programming language.
- It has 32kB of RAM/program memory and 32kB of EEPROM.
- It operates at 5VDC and can sink/source 30mA of current per I/O pin.
- It operates at a speed of 20MHz.
- The Javelin has built in Virtual Peripherals (VPs) that take care of serial communication, pulse width modulation and tracking time in the background.
- The Javelin Stamp Integrated Development Environment (Javelin Stamp IDE) software is used with the Javelin connected to a PC by a serial cable.
- It supports delta-sigma analog to digital conversion.
- D/A conversions are accomplished in the background as a continuous pulse train delivered by an I/O pin.

### *Fixed point arithmetic*

Javelin stamp processor does not support floating point operations. However, applications like Kalman filter based adaptive sampling and localization using dead reckoning requires floating point computations. In order to achieve this, a fixed point approach is used to represent a floating point number.

A fixed-point number  $n$  may be written as

$$n = m.f$$

Where,  $m$  = the magnitude, i.e. the integer part,

"." = the radix point,

$f$  = the fractional part.

The integer part can be represented as a power of 2, while the fractional bit is represented as inverse power of two.

The binary representation of a floating point number is shown below

Table 3.1 Fixed point representation of a floating point number

Integer (m)						Radix point	Fractional part (f)				
...	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	.	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	...

We have currently implemented mathematical operations such as addition, subtraction, multiplication and division using fixed point method.

#### *PIC microcontroller*

PIC 12F508 is used to supply PWM pulses continuously to the two RC servo motors. The PIC takes away the PWM generation onus from the Javelin through a single pin interface with the stamp.

An extensive assembly language coding is worked out to control the speed and directions of the motors.

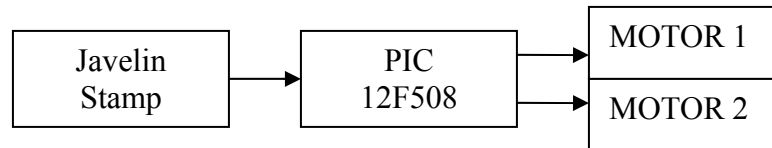


Figure 3.3 Block diagram of motor control

### *Servos on ARRI BOT*

There are two Futaba's S148 RC servos (Radio controlled) on ARRI BOT. The servo module has a built in motor, gearbox and the controlling electronics [35].

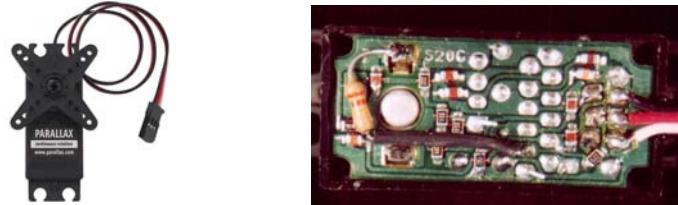


Figure 3.4 Picture of Futaba's S148 servo and internal circuitry of the servo

The direction and speed of the motors is controlled through PWM pulses generated by PIC 12F508 microcontroller. The technical specification of the motor is given below.

Table 3.2 Technical specification of the servos

<b>Parameter</b>	<b>Value</b>
Power	6v DC max
Average speed at 5v	60rpm
Torque	3.40 kg-cm/47oz-in
Size (L x W x H) inches	1.60 x .79x 1.50

### *Optical Encoders on ARRI BOT*

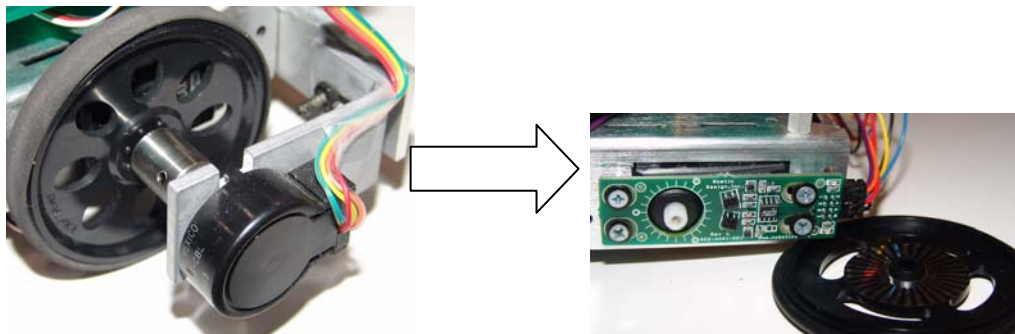


Figure 3.5 Picture of the first generation optical encoders and the second generation optical encoders.

The new encoders (WW-01) from Nubotics are mounted on the servos to sense the direction and distance traveled by the motors. In comparison to the old encoders 600EN-128-CBL on the first generation of ARRI BOT, the new WW-01 are easy to mount, can be directly interfaced with the processor and also cuts the cost by half.

*Wheel resolution achieved on ARRI BOT*

Highly precise maneuverability of resolution less than 3mm is achieved through continuous servo motors and optical encoders.

Encoder Resolution = 64 pulses per rotation

Wheel Diameter = 2.75 inches

Wheel Circumference = 8.639inches

Distance traveled in one rotation~21.9cm

$$\text{Resolution Achieved} = \frac{21.9}{64} = 0.342\text{cm} \quad \text{i.e. } \sim 3\text{mm}$$

By amplifying the encoder resolution to 128 pulses per rotation a small motion up to 1.7mm can be captured.

*Color Sensor*

ARRI Bot first and second generations are equipped with a TAOS TCS230 color sensor to validate deployment algorithms such as adaptive sampling based on the color information obtained from a linear parametric, color test bed.



Figure 3.6 TCS230 Color Sensor Evaluation Kit [49]

The color sensor module is comprised of a color detector including a TAOS TCS230 RGB sensor chip, white LED and a collimator lens. It has an array of photo detectors each with a red, green, or blue filter or no filter (clear). The filters are distributed evenly throughout the array to eliminate location bias among the colors

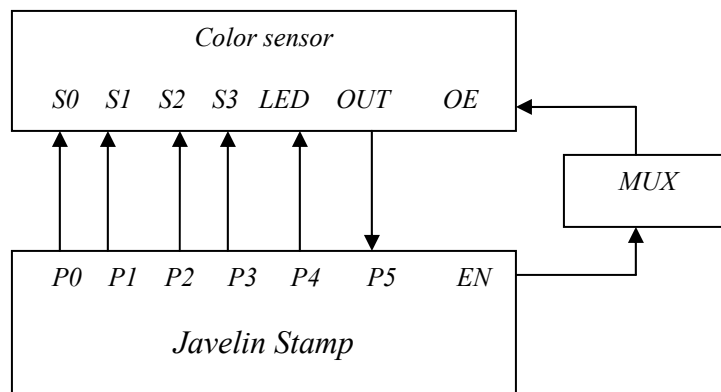


Figure 3.7 Block diagram of color sensor interface with CPU

Internal to the device is an oscillator which produces a square-wave output whose frequency is proportional to the intensity of the chosen color. There is one tri-stable output from this oscillator, and the color to be read is selected using two address lines S2 and S3. In addition, it is possible to program the divide rate of the oscillator

using S0 and S1 lines. When an output is to be read, the processor enables the color sensor through a multiplexer as shown in the block diagram. The sensor module's OE (output enable) line further gates the LED enable, so that the board's LEDs cannot be on if the output is disabled. This helps in preventing the contention on the line if two sensor modules are connected and to prevent excess current drain through the two sets of LEDs being on concurrently [49].

### *Ultrasonics on ARRIBOT*

There are two ultrasonic range finder used on ARRIBOT (SRF04) as shown in figure 3 for obstacle avoidance during navigation.



Figure3.8 Ultrasonic Transducer (SRF04)

Table 3.3 Properties of Ultrasonic Transducer

Parameter	Value
Beam Pattern	Shown in figure 3
Voltage	5v
Current	30mA Type. 50mA Max
Frequency	40KHz
Maximum Range	3 m
Minimum Range	3 cm
Sensitivity	Detect a 3cm diameter stick at > 2 m
Input Trigger	10uS Min. TTL level pulse
Echo Pulse	Positive TTL level signal, width proportional to range.
Weight	0.4 oz.
Size	1.75" w x 0.625" h x 0.5" d

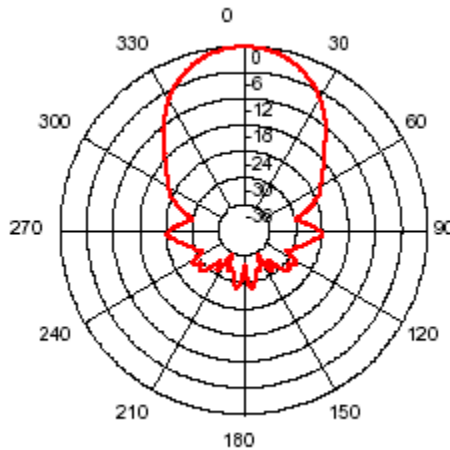


Figure 3.9 Beam Pattern of the Ultrasonic Transducer [50]

*Beam Pattern of Ultrasonic transducers*

Beam pattern is defined as the sensitivity of the transducer as a function of spatial angle. This pattern can be determined from the frequency of operation and the size, shape and acoustic phase characteristics of the vibrating surface [37]. The



narrowness of the beam pattern for a circular radiating surface vibrating in phase is a function of the ratio of the diameter of the radiating surface to the wavelength of sound at the operating frequency. The larger the diameter of the transducer as compared to a wavelength of sound, the narrower the sound beam will be.

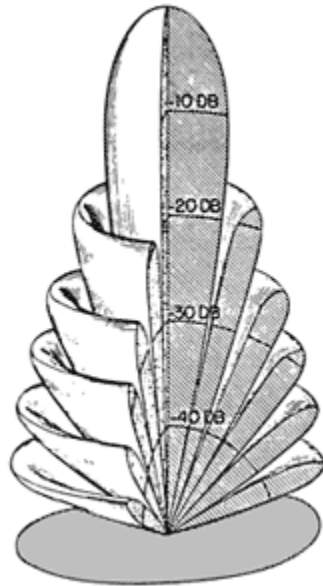


Figure 3.10 3-D representation of the beam pattern [37]

As can be seen, it produces a narrow conical beam and a number of secondary lobes of reduced amplitude separated by nulls. The beam angle of a transducer is usually defined as the measurement of the total angle where the sound pressure level of the main beam has been reduced by 3 dB on both sides of the on-axis peak. As seen in the figure the transducer still has the sensitivity at greater angles, both in the main beam and in the secondary lobes.

### *Ultrasonic Interface with CPU*

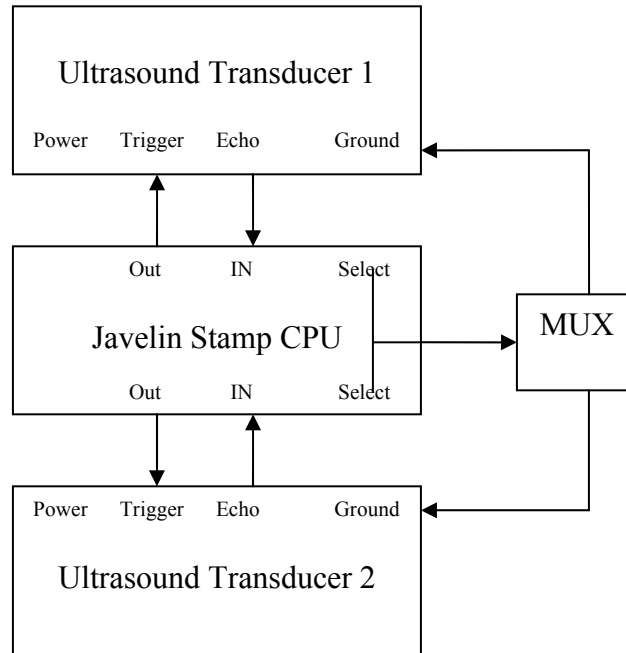


Figure 3.11 Block diagram of ultrasonic interface to CPU

### *Operation*

The CPU sends out a short 10 $\mu$ s pulse trigger to the transducer for ranging. The SRF04 on receiving the trigger sends out an 8 cycle burst of ultrasound at 40 kHz and raise its echo line high. It then listens for an echo, and as soon as it detects one it lowers the echo line. The transducer will acknowledge only the first echo received and ignore the rest in a multi-echo scenario. The echo line is now a pulse whose width is proportional to the distance to the object. By timing the pulse it is possible to calculate the range in inches/centimeters or anything else. If nothing is detected then the SRF04

will lower its echo line anyway after about 36mS. The SRF04 Timing diagram is shown below in figure 21.

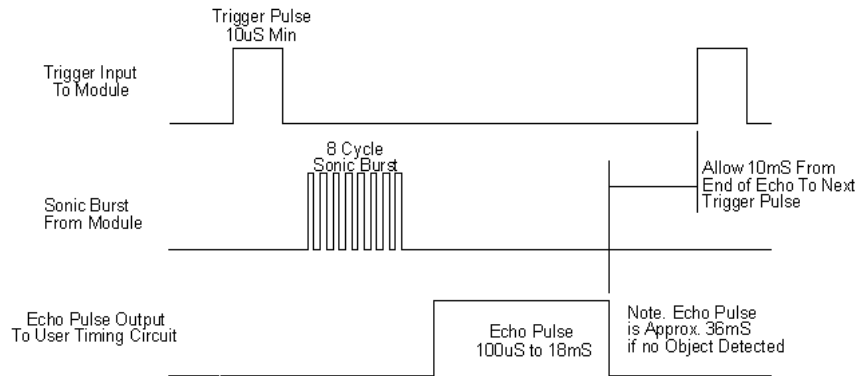


Figure 3.12 Timing Diagram of the Ultrasonic Transducer (SRF04) [38]

### *Range Calculation*

The SRF04 provides an echo pulse proportional to distance. If the width of the pulse is measured in  $\mu\text{S}$ , then division by 58 gives the distance in cm, otherwise division by 148 gives the distance in inches.

### *Alternative Energy Source Exploration*

With the increasing amount of robotic applications, the robots face various challenges one of which is related to power driving the robots. Powering the robots is the very first and crucial step in all designs. Harvesting an alternative source of energy to recharge the batteries on the robots is a well suited solution for autonomous application.

With the advancement of technology there are various methods of converting naturally available source to energy to electrical or mechanical energy, some of which are mentioned below.

- Solar Cells
- Thermocouples
- Wind Turbines
- Vibration

#### *Solar Cells*

The sun has always been the most reliable, free and limitless source of energy and is expected to last for at least for next five billion years or so. Research in solar energy has explored various methods to harvest it. The sunlight is a combination of several electromagnetic waves of different wavelengths including the Ultraviolet UV ('above violet', 290 nm, and short and high energy wavelength) to visible light wavelengths (400nm to 760nm) to Infrared radiations ('below red', 3000 nm , low energy wavelength)

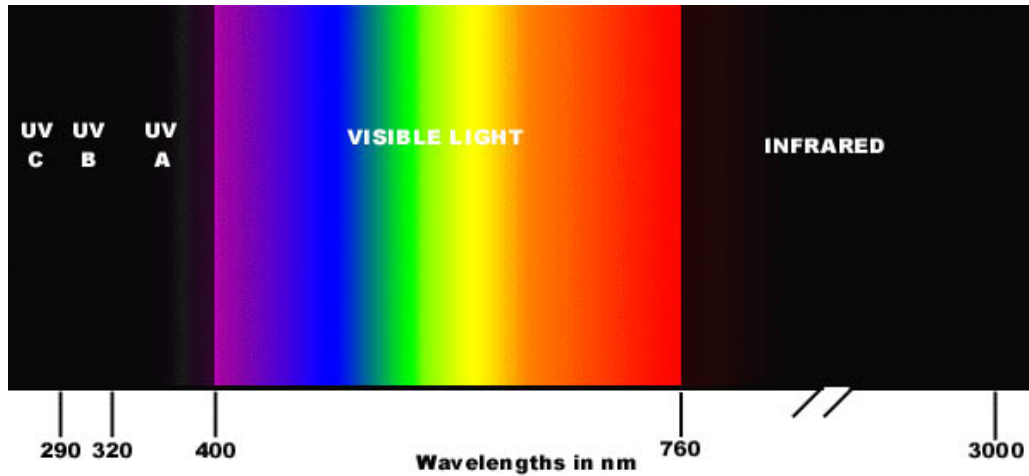


Figure 3.13 Spectrum of Sunlight [56]

Solar panels are made of photovoltaic cells that convert sunlight to electrical energy. The most commonly used photovoltaic PV cells are made of silicon semiconductor. When sunlight strikes the cells the light energy is transferred to the electrons which set them free and generates electron flow i.e. current. The PV cells have some kind of electric field that controls the direction of current flow. By connecting metal contacts on the top and bottom of the cells, current is drawn as an output. This current along with the cell's voltage which is the resultant of the built in electric field defines the power of a solar panel. The efficiency of the solar cell varies based on the material used to manufacture it.

Table 3.4 Efficiency achieved with different types of solar cells

Type of solar cell	Efficiency %
Single-crystal silicon	14 to 17
Polycrystalline silicon	5 to 7
Copper indium diselenide	8 to 10

### *Solar Cells on ARRI Bot*

The ARRI Bot has been equipped with a Plastecs's solar panel (shown in figure 6) which is capable of providing a power of 2.5~3 Watts at a peak current of 200mA (SC). The panel was selected on the basis of a tradeoff between the size and its output capacity. The size of the solar panel was selected small enough to fit on the robot while providing a sufficient output to charge a 9V battery.

Table 3.5 Specification of solar panel on the ARRI BOT

Open circuit voltage	15-17 volts
Short circuit current	200mA
Power output	2.5 to 3 Watts
Size	4.5 x 8.5 inches

This panel, along with a battery charging circuit, helps to harvest hand-on energy required for the robot and also to recharge the conventional battery power supply.



Figure 3.14 Solar Panel of ARRIBOT [57]

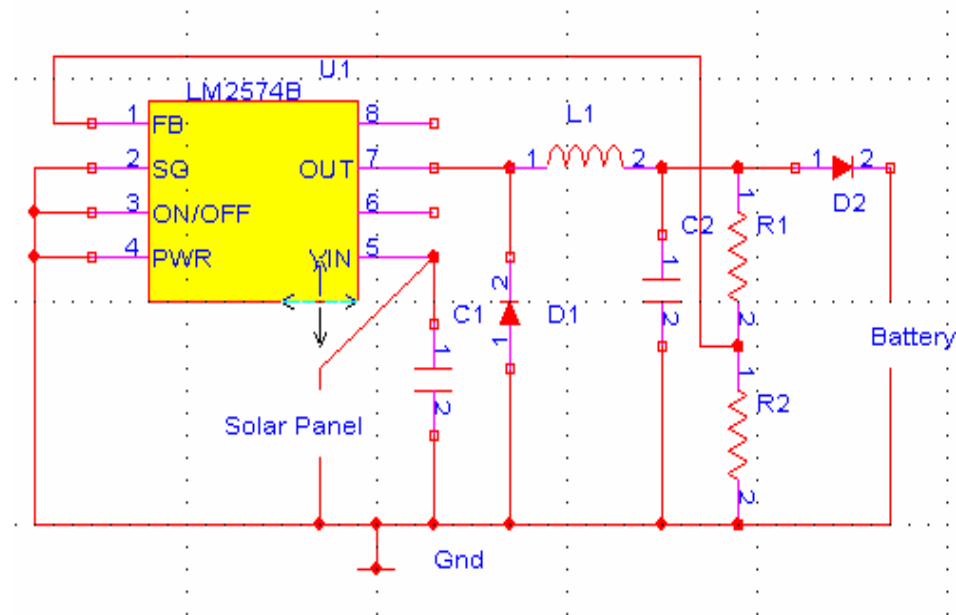


Figure 3.15 Circuit for solar energy management

### *Circuit Description*

The energy harvested from the solar panel should be efficiently relinquished to the rechargeable batteries in order to steer the robots. This is achieved by means of a switch- mode step- down converter (LM2574) (given in Figure 8). The ratio of solar panel no-load voltage to battery voltage determines the choice of converter. A step up converter also called as Boost converter is used if the solar panel no load voltage is lower than the voltage of the discharged battery, while a step down converter (Buck converter) is used if the ratio of the panel voltage under load is higher than that of the fully charged battery. To ensure that every bit of energy obtained from the solar is efficiently stored in the battery, the switch- mode step down converter executes two cycle of operation, in the first cycle it connects the inductor (L1) to the power source

allowing a buildup of energy in the inductor (L1). In the second cycle, a change in current path enables the inductor to transfer the accumulated energy to the rechargeable batteries.

To cope up with the continuously varying current output from the solar panel a reservoir capacitor C1 is used to store the output of the solar panel till it approaches an adequate value to start the converter. When the capacitor voltage falls below the operating voltage of the converter, it will switch off the converter until the capacitor again charges to the optimum voltage.

A diode D2 is connected at the output to block reverse flow of current when the battery voltage is higher than the output of solar panel.

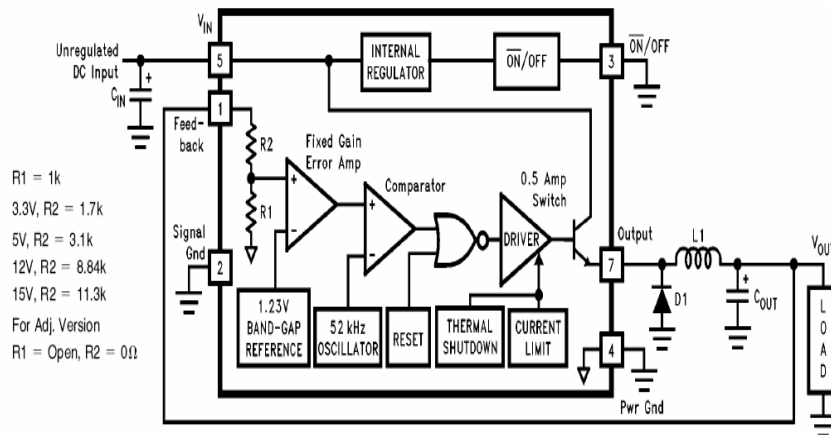


Figure 3.16 Block Diagram of the step- down converter (LM2574) [51]



### *Designing of the Circuit*

$V_{in}(\text{max})$  = Maximum input voltage

$V_{out}$  = Regulated output voltage

$I_{load}(\text{Max})$  = Maximum load current

F = Switching Frequency (Fixed at 52 KHz)

#### *— Selecting resistor values*

$$V_{out} = V_{ref} \left(1 + \frac{R_1}{R_2}\right), V_{ref} = 1.23V$$

$R_2$  can be between 1K and 5K,

$$R_1 = R_2 \left(\frac{V_{out}}{V_{ref}} - 1\right)$$

#### *— Selecting Inductor values*

$$E \bullet T (V \bullet \mu s)$$

$$E \bullet T = (V_{in} - V_{out}) \frac{V_{out}}{V_{in}} \bullet \frac{1000}{F(\text{KHz})} (V \bullet \mu s)$$

- The  $E \bullet T$  value is used to match with the  $E \bullet T$  number on the vertical axis of the inductor selection guide shown in the figure 26.

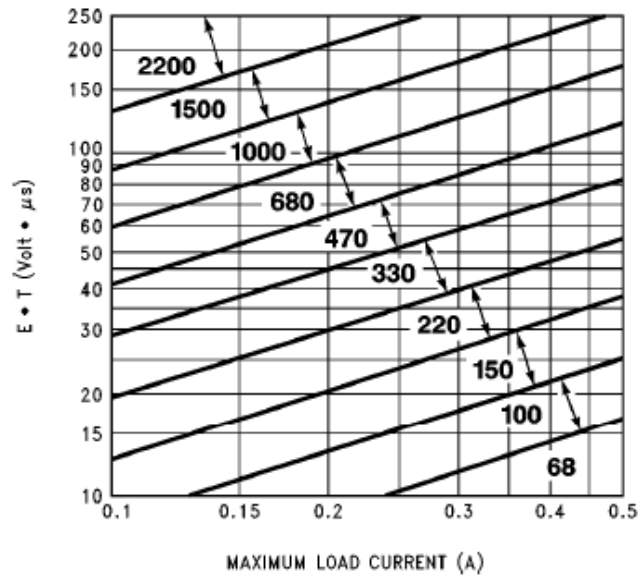


Figure 3.17 Characteristic Curves for the Inductor Selection [51]

- Maximum load current is selected along the horizontal axis.
- The intersection of  $E \bullet T$  and maximum load current identifies the inductance region and the inductor value is selected.
- The inductor chosen must be rated for operation at the LM2574 switching frequency (52 KHz) and for a current rating of  $1.5 \times I_{load}$ .

--- *Selecting Output Capacitor*

The value of output capacitor along with the inductor value determines the pole-pair of the regulator, for a stable operation the capacitor should satisfy the following equation.

$$C_2 \geq 13,300 \frac{V_{in}(\max)}{V_{out} \bullet L(\mu H)} (\mu F)$$



### *Piezo transducers*

Piezoelectric transducers are used to convert the vibration energy generated by a moving robot to electric energy in order to charge the on-board batteries. A piezo panel was designed by another research group at ARRI for this purpose.

The panel consist of three sets of piezo transducers to capture vibration along the x, y and z directions. A rectifier circuit is used to convert the a.c output from the transducer to a d.c voltage required to charge the batteries.

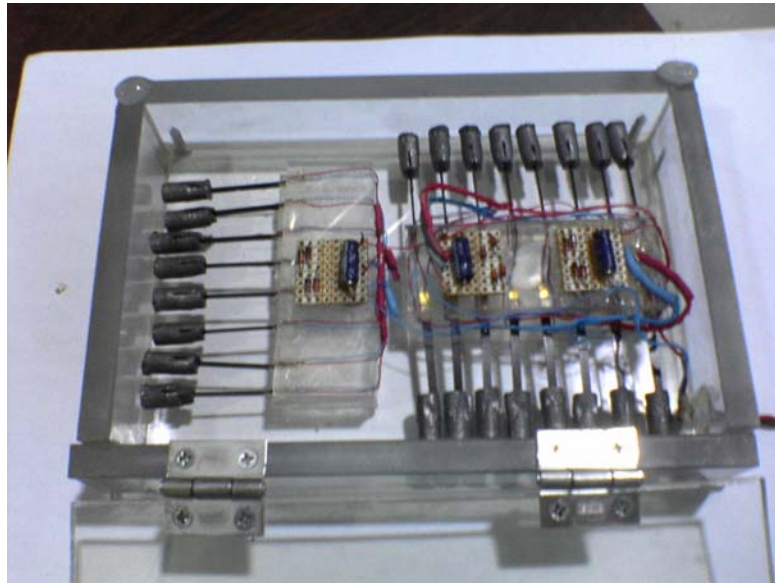


Figure 3.19 Piezoelectric panel

### *Wireless Communication through Cricket Sensor Nodes*

The second generation ARRI Bot design has replaced the Parallax's wireless communication module (part # 27988) with Crossbow's motes.



Figure 3.20 Parallax card [64], MICA2 [53], Cricket [52]

### *MICA2 Wireless communication module*

MICA2 is Crossbow's third generation, microcontroller based, low power wireless sensor node. A multi-channel transceiver (868/916 MHz, 433 MHz and 315 MHz) and programmable range helps in cautious utilization of energy based on applications [53].

NesC is used to program the motes through Tiny OS v1.0 platform. Tiny OS 1.0 is a small, open-source, energy efficient, software operating system developed by UC Berkeley which supports large scale, self configuring sensor networks. The processor and radio module MPR400CB is based on Atmel's ATmega128L which runs TOS from its flash memory. MICA2 also offers a variety of sensors and data acquisition boards like the Light, Temperature, RH, Barometric, Pressure, Acceleration/Seismic, Acoustic, Magnetic and others through a standard HIROSE's 51 pin connector.

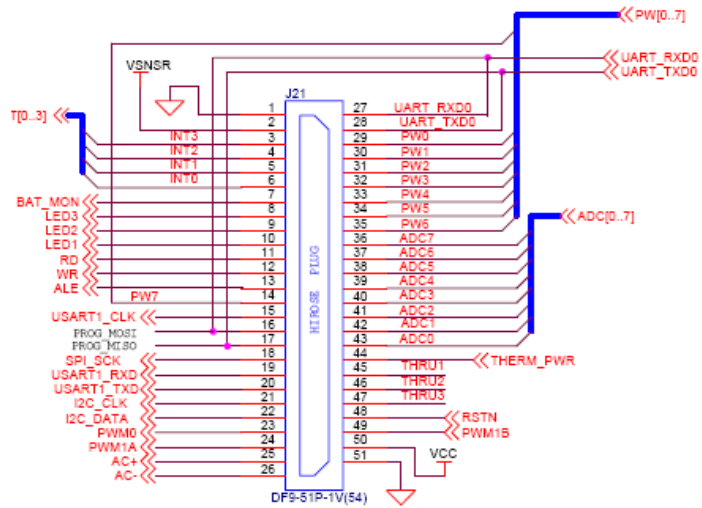
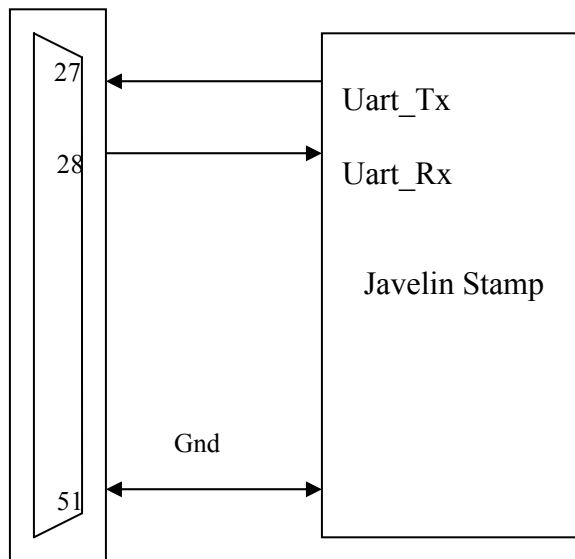


Figure 3.21 MICA2's 51 pin connector [53]



MICA2 51 Pin connector

Encoder

Figure 3.22 MICA2 and ARRIBOT interface

## *Crossbow's Crickets*

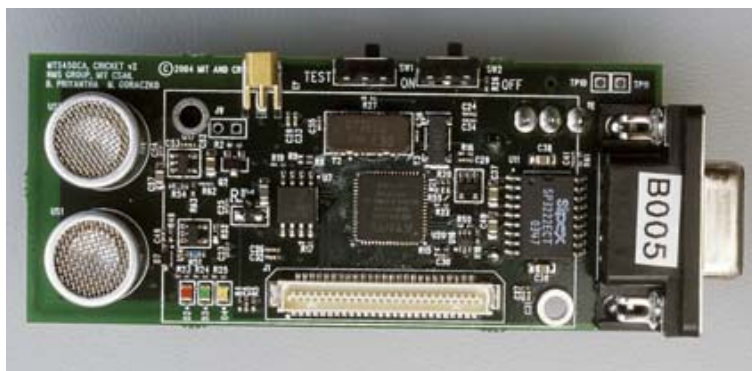


Figure 3.23 Cricket Sensor Node [52]

Cricket v2 are the second generation crossbow motes and are substantially more accurate and energy efficient than the first generation crickets. It is a location-aware MICA2 Processor/Radio module that operates using a combination of Ultrasound and RF signals. Cricket is envisioned for indoors applications or in urban areas where outdoor systems like the Global Positioning System (GPS) don't work well. Its radio runs at a frequency of 433 MHz, with the default transmit power level and antennas providing a range of about 30 meters indoors when there are no obstacles. The maximum ultrasound range is 10.5 meters when the listener and the beacon are facing each other and there are no obstacles between them. It can provide distance ranging and positioning information with a precision of 1 to 3 cm off from the actual distance.

### *Principle of Operation*

Cricket operates in three modes, namely the Radio, Beacon and the Listener mode [52]. Cricket uses a combination of RF and ultrasound technologies to provide location information to attach host devices. Static beacons publish information on an RF

channel followed by a concurrent ultrasonic pulse. Mobile or static attached to devices listen for RF signals, and upon receipt of the first few bits, listen for the corresponding ultrasonic pulse. When this pulse arrives, the listener obtains a distance estimate for the corresponding beacon by taking advantage of the difference in propagation speeds between RF (speed of light) and ultrasound (speed of sound). The listener runs algorithms that correlate RF and ultrasound samples (the latter are simple pulses with no data encoded on them) and to pick the best correlation. Even in the presence of several competing beacon transmissions, Cricket achieves good precision and accuracy quickly.

#### *Interfacing the Cricket Sensor Node with ARRI BOT*

The cricket provides data on its serial port and communicates with the host device using RS 232 serial connection. For this reason, the crickets are coupled to ARRI BOT through RS 232 interface.

The RS 232 (Recommended Standard) protocol was developed by EIA/TIA (Electronic Industry Association and Telecommunication Industry Association) and is defined as “Interface between Data Terminal Equipment and Data Circuit-Termination Equipment Employing Serial Binary Data Interchange.” The standard builds compatibility between the host and the peripheral system by specifying information such as

1. Common voltage and signal levels.
2. Common pin-wiring configurations.



3. A minimal amount of control information between the host and peripheral systems.

Table 3.6 RS-232 Specifications

<b>Parameter</b>	<b>Specification</b>
Cabling	Single-ended
Number of Devices	1 transmit, 1 receive
Communication Mode	Full duplex
Distance (max)	50 feet at 19.2kbps
Data Rate (max)	1Mbps
Signaling	Unbalanced
Mark (Logic 1)	-3V (min) -12V (max)
Space (Logic 0)	3V (min) 12V (max)
Input Level (min)	±3V
Output Current	500mA (Note that the driver ICs normally used in PCs are limited to 10mA)
Impedance	5k $\Omega$ (Internal)
Bus Architecture	Point-to-Point

The RS232 standard specifies a DB25 pin connector for serial communication however the most popular one is DB9 pin connector which includes the necessary signal to build serial communication between the transmitter and receiver. Table shows the pin out for 25 and 9 pin connector.

Table 3.7 RS 232 DB25 pin description



	<b>25 Pin Connector on a DTE device (PC connection)</b>
Male RS232 DB25	
<b>Pin Number</b>	<b>Direction of signal:</b>
1	Protective Ground
2	Transmitted Data (TD) Outgoing Data (from a DTE to a DCE)
3	Received Data (RD) Incoming Data (from a DCE to a DTE)
4	Request To Send (RTS) Outgoing flow control signal controlled by DTE
5	Clear To Send (CTS) Incoming flow control signal controlled by DCE
6	Data Set Ready (DSR) Incoming handshaking signal controlled by DCE
7	Signal Ground Common reference voltage
8	Carrier Detect (CD) Incoming signal from a modem
20	Data Terminal Ready (DTR) Outgoing handshaking signal controlled by DTE
22	Ring Indicator (RI) Incoming signal from a modem

Table 3.8 RS 232 DB9 pin description

	<b>9 Pin Connector on a DTE device (PC connection)</b>
Male RS232 DB9	
<b>Pin Number</b>	<b>Direction of signal:</b>
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) Outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

Since the standard assigns -3V to -12V to Logic 1 and 3V to 12 V to logic 0, there is need for a logic converter (MAX232 chip) to make it compatible with the TTL/CMOS signal levels. The interface between the cricket and ARRI BOT through MAX232 is shown in the following figure.

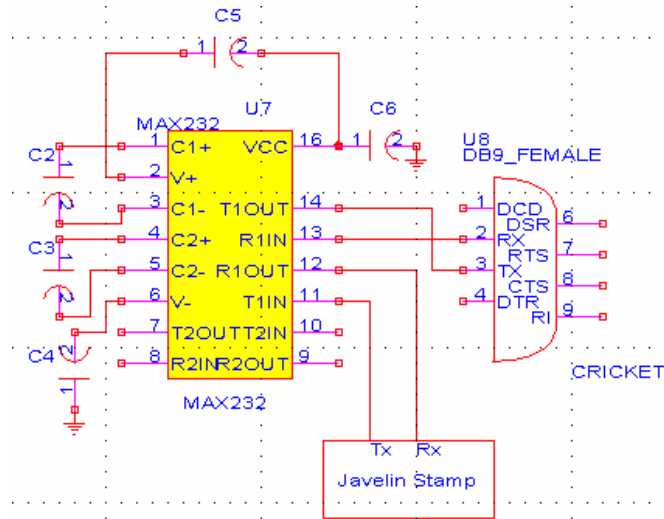


Figure 3.24 RS232 Communication Interface

The capacitors are used as charge pumps, C1 doubles the +5v to +10v on C3 at V+ output and C2 inverts the +10V to -10V on C4 at the V- output.

*Power Budget*

The power consumed by each module on the robot is listed in descending order in the table below. The power indicated in the table, is consumed by each module at the time of its operation. Generally, only one module is selected at a given time, expect for the Javelin stamp, optical encoders, cricket, MAX 232 and the multiplexer which operates continuously. The motors, ranging module, counter and the color sensor consume power only when they are selected by the Javelin stamp to operate. Therefore,

the total power computed in the table, represents the maximum power that would be required if all the modules were to be switched on at a given time.

Table 3.9 Power Consumption budget of the modules on ARRI BOT

Modules	Current	Power
Motors (2)	160mA	.8 W
Javelin Stamp	120mA	.6 W
Ultrasonic Range Finder(2)	100mA	.5 W
Optical Encoder(2)	60mA	.3 W
Cricket (Listener)	33mA	.165W
Clock	30mA	.15 W
Counter	30mA	.15 W
MAX 232 IC	15mA	0.075 W
MUX	4mA	0.02 W
Color Sensor	3mA	0.015W
<b>TOTAL</b>	<b>555 mA</b>	<b>2.775W</b>

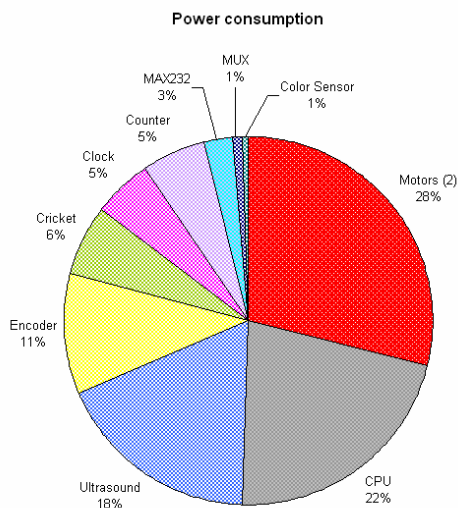


Figure 3.25 Power Distribution

With a 6V 4000mAh NiMH rechargeable battery, the robot can operate for approximately for 8 hours.

### PCB DESIGN

A customized PCB is designed to accommodate the electronic modules on ARRI BOT using PCB123 software. PCB123 software is offered by Sunstone Circuits [58] to design PCB layouts and to place manufacturing order online.

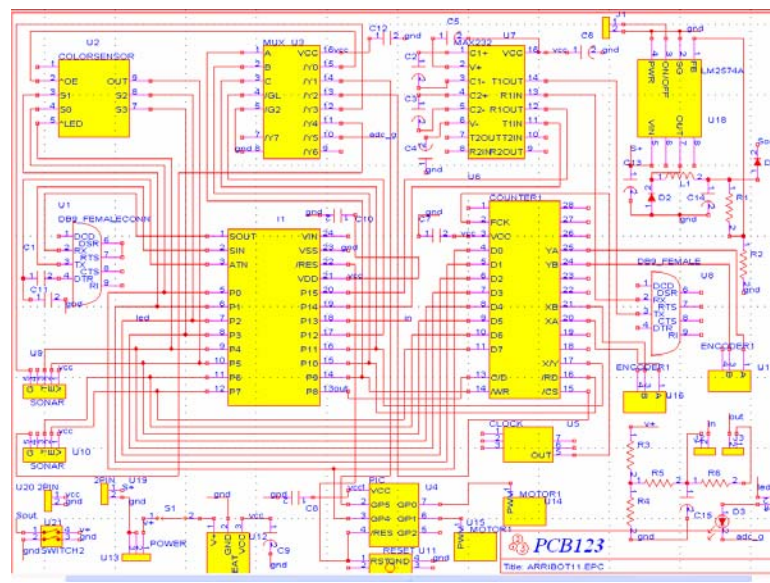


Figure 3.26 Schematic diagram of ARRI BOT

A double sided PCB has been used to achieve a compact design.

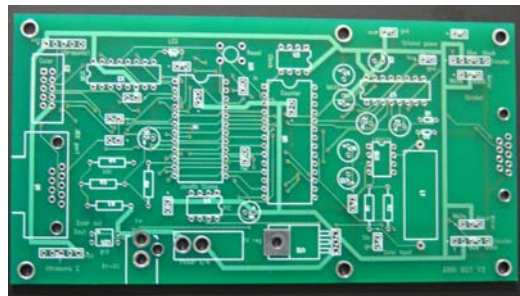


Figure 3.27 PCB layout of ARRI BOT

### 3.2 Structural design of ARRI BOT

A new chassis was designed for the upgraded ARRI BOT, to house a new PCB, solar panel, ultrasounds, color sensor, crickets, the battery pack and Omni directional front wheel.

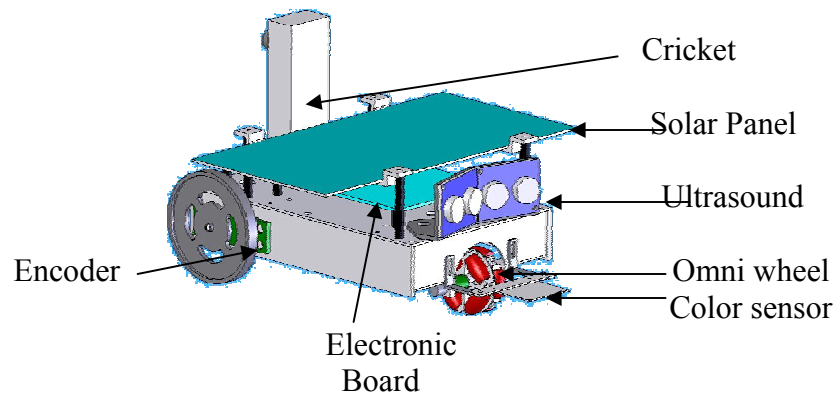


Figure 3.28 Structural Model of ARRIBOT

The chassis parts were modeled using Solid Works®. Some of the constraints that were addressed while designing the chassis are as follow

- The crickets should be centered between the two back wheels along the axle of the servos for localization purposes.
- The color sensor should be mounted in way to maintain one inch spacing from the ground level.
- The placement of ultrasound should be optimized to cover a broad area.

*Mounting criteria for the ultrasonic modules*

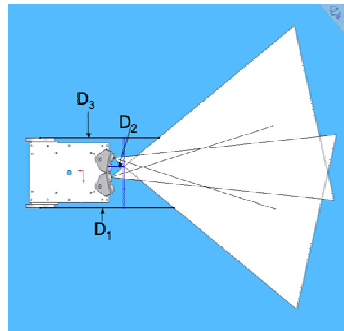


Figure 3.29 Optimized mounting of the ultrasonic modules

As shown in the figure, D1 and D3 are the distances from the right and left wheel to the point where it intersects the cone formed by the two ultrasonic modules. D2 is the distance which is not covered by either of the ultrasonic modules, which means that if an obstacle is placed within this distance in the center of the two ultrasonic, then the object will not be detected. The table shows different combination computed to achieve an optimized placement of the ranging modules. The best combination selected offers a minimum D2 value and optimized values for D1 and D3 distances.

Table 3.10 Different mounting combination of ultrasonic modules

Distance between the ultrasonic	Placement angle	D1 (inch)	D2 (inch)	D3 (inch)	Total distance (inch)
1.97	0°	13.68	3.0	11.48	28.34
1.97	15° out	10	15.06	8.73	33.79
<b>2.17</b>	<b>15° in</b>	<b>13</b>	<b>1.23</b>	<b>11.63</b>	<b>26.16</b>
2.17	0°	13.49	3.35	11.25	28.09
2.17	15° out	9.93	16.14	8.56	34.63
2.36	15° in	13.14	1.66	11.72	26.52
2.36	0°	13.38	3.61	11.02	28.01
2.36	15° out	9.76	17.31	8.4	35.47

### 3.3 Software control for ARRI BOT

Javelin Stamp Integrated Development Environment (IDE) version 2.0.3 is used to program the robots. A structured and object oriented programs have been written to enable application based modifications and flexible programming.

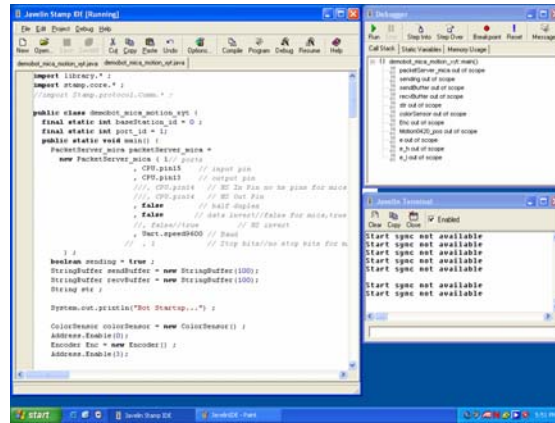


Figure 3.30 Snap shot of Javelin Stamp IDE

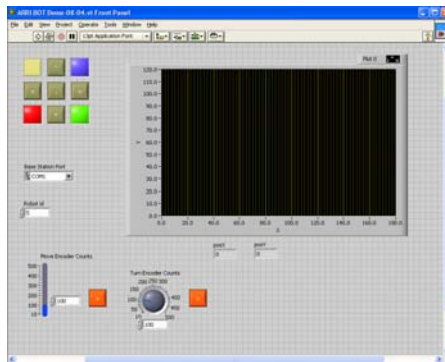


Figure 3.31 Snap shot of Labview based user interface

As shown in the figure, a Labview® based user interface is designed to establish communication between the robot and the base station. Commands from the base station can be either addressed to a single robot or to multiple robots through the interface. The plotting space at the top right corner is provided to mark the position of the robots during localization.



*Software control*

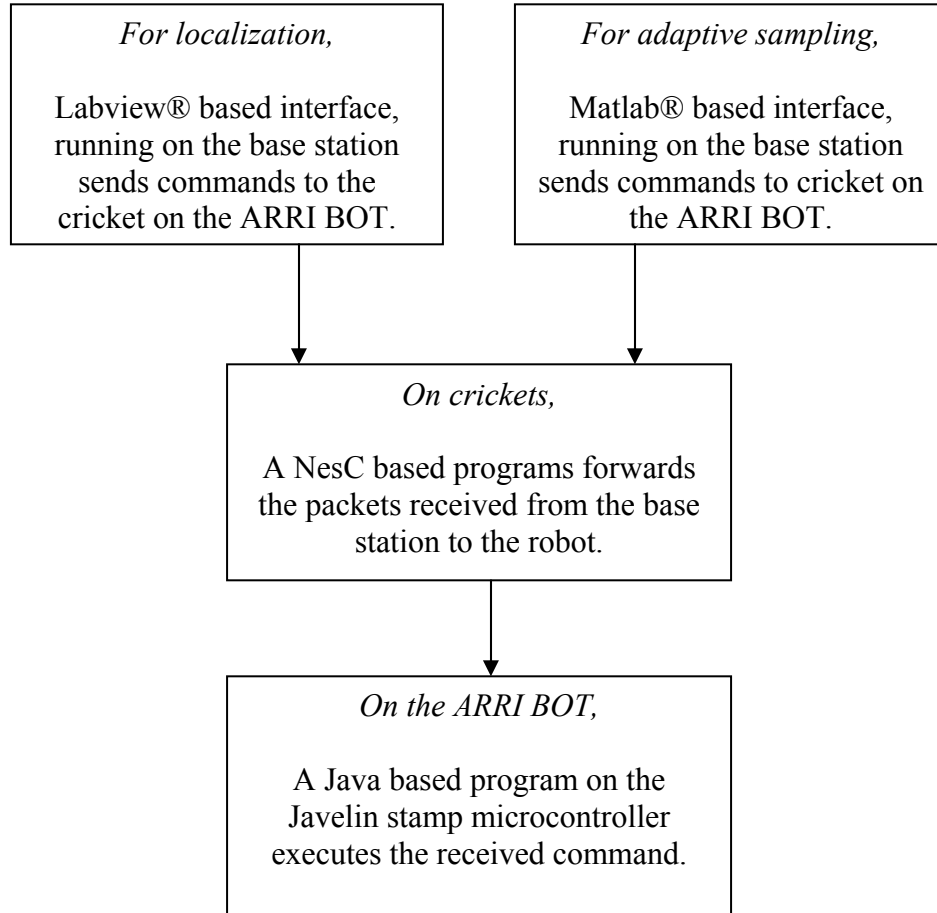


Figure 3.32 Flowchart of software controls.

*List of Commands*

Table 3.11 List of software command for ARRI BOT

<b>Commands</b>	<b>Interpretation</b>
'F'	Move forward infinitely
'M250'	Move forward by the specified number of encoder values for e.g. move for 250 encoder value, where $0 < \text{enc value} \leq 32767$
'B'	Move Backward infinitely
'M-250'	Move Backward by the specified encoder value for e.g. move backward for 250 encoder value, where $-32767 \leq \text{enc value} < 0$
'S'	Stop
'L'	Turn Left infinitely
'T20'	Turn Left for the specified encoder value $0 < \text{encoder value} \leq 32767$
'R'	Turn Right infinitely
'T-20'	Turn Right for the specified encoder value $-32767 \leq \text{encoder value} < 0$
'C'	Take a color sample
'J'	Switch the crickets mode from radio to Beacon for 15 sec and then back to radio
'Z'	Switch the crickets mode from radio to listener for 10 sec and then back to radio

## CHAPTER 4

### TOOLS AND ALGORITHMS FOR DEPLOYMENT

The previous chapter discussed the features of the on-board modules of ARRI BOT. This chapter discusses some of the currently researched deployment algorithms. ARRI BOT is a research platform and is a potential contender for testing various robotic concepts like localization, path planning, navigation, adaptive sampling etc. In this chapter, we discuss how deployment algorithm like the basic wireless communication protocol, localization and adaptive sampling are implemented on ARRI BOT.

#### 4.1 Wireless Communication on ARRI BOT

The second generation ARRI BOT were first tested with Crossbow's MICA2 to establish wireless communication and then replaced with Crickets to achieve localization along with wireless communication.

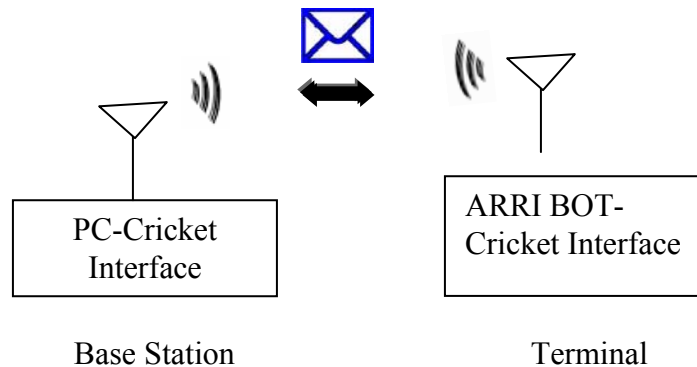


Figure 4.1 Block diagram of wireless communication between base station and ARRI BOT

With MICA2 and Crickets, data is transmitted or received based on TinyOS [17] serial data format. Some of the key points in forming or deciphering a TinyOS serial data packet are as follows,

- A TinyOS data packet has a maximum length of 255 bytes.
- The raw data packet is wrapped on both ends by a frame synchronization byte of 0x7E. This is used to detect the start and end of a packet from the stream.
- The raw data packet uses an escape byte of **0x7D**. This is needed in case a byte of payload data is the same as a reserved byte code, such as the frame synch byte **0x7E**. In such a case, the payload data will be preceded by the escape byte and the payload data itself will be exclusively OR'ed with **0x20**. For example, a payload data byte of **0x7E** would appear in the data packet as **0x7D 0x5E**.
- On an XP machine, multiple byte values are byte-swapped in the data stream. For example, the 2 byte UART Address field (0x007E) will appear as 7E 00 in the byte stream. (Note: the .NET BitConverter.ToUInt16 () handles this conversion correctly).

Table 4.1 A typical TinyOS packet

Address		Message type	Group ID	Data Length	Payload data	CRC	
0	1	2	3	4	5...n-2	n-1	n

### *Outlining Payload Data*

A robust communication is engineered by including robot ids, network id, sequence number, CRC check and acknowledgement schemes in the payload data slot of TinyOS serial data packet. The construction of the payload data is carried out using MATLAB® at the base station and using Javelin Stamp on the robots.

### *A typical Payload data Packet*

A typical payload data packet for a robust communication protocol is shown below

Table 4.2 A typical Payload data Packet

<i>Length Of the data</i>	<i>Sequence Number</i>	<i>Network ID</i>	<i>Robot ID</i>	<i>Command</i>	<i>CRC Lower Byte</i>	<i>CRC Higher Byte</i>
0x07	0x01	0xFF	0x01	'M100'	0xFF	0xFF

### *Length of payload data*

Length of payload data represents the total number data bytes transmitted in a TinyOS packet. It does not include the CRC or frame sync bytes. The basic TinyOS packet can transmit up to 29 data bytes (excluding CRC bytes) in a single packet.

### *Sequence Number*

Sequence Number was added for the purpose of Acknowledgement retransmission scheme to achieve a robust communication protocol. The ARRI BOT and the Base station keep a record of the number of packets transmitted with the help of the Sequence number. On receiving a packet it sends back an acknowledgement packet with the same sequence number as that of the received packet.

### *Network ID*

Network Id was added to enable the robot and the base station to process data/command packets meant for ARRI Bots MSWN only and ignore others in a multi-network scenario.

### *Robot ID*

The robot id helps to control either a single robot (Uni-casting) or a set of robots (Multicasting) or to all the robots on the test bed (Broadcasting).

### *Command or Data bytes*

This field holds either commands for the robot or sensor data information or acknowledgement information.

### *CRC Bytes*

The transmitter performs a two byte CRC-CCITT calculation on the raw data and appends the result to the packet. At the receiver end CRC calculation is performed again and the value is compared with the transmitted value, if there is a match then a valid packet is received else the garbled packet is discarded. This helps in achieving an error free communication.

### *TinyOS serial data packet formation*

For MICA2, packet formation is done in MATLAB® at the base station and on JAVELIN STAMP on the robot, where as, the Crickets take care of the packet formation independently. The steps for a TinyOS serial data packet formation is shown as follows;

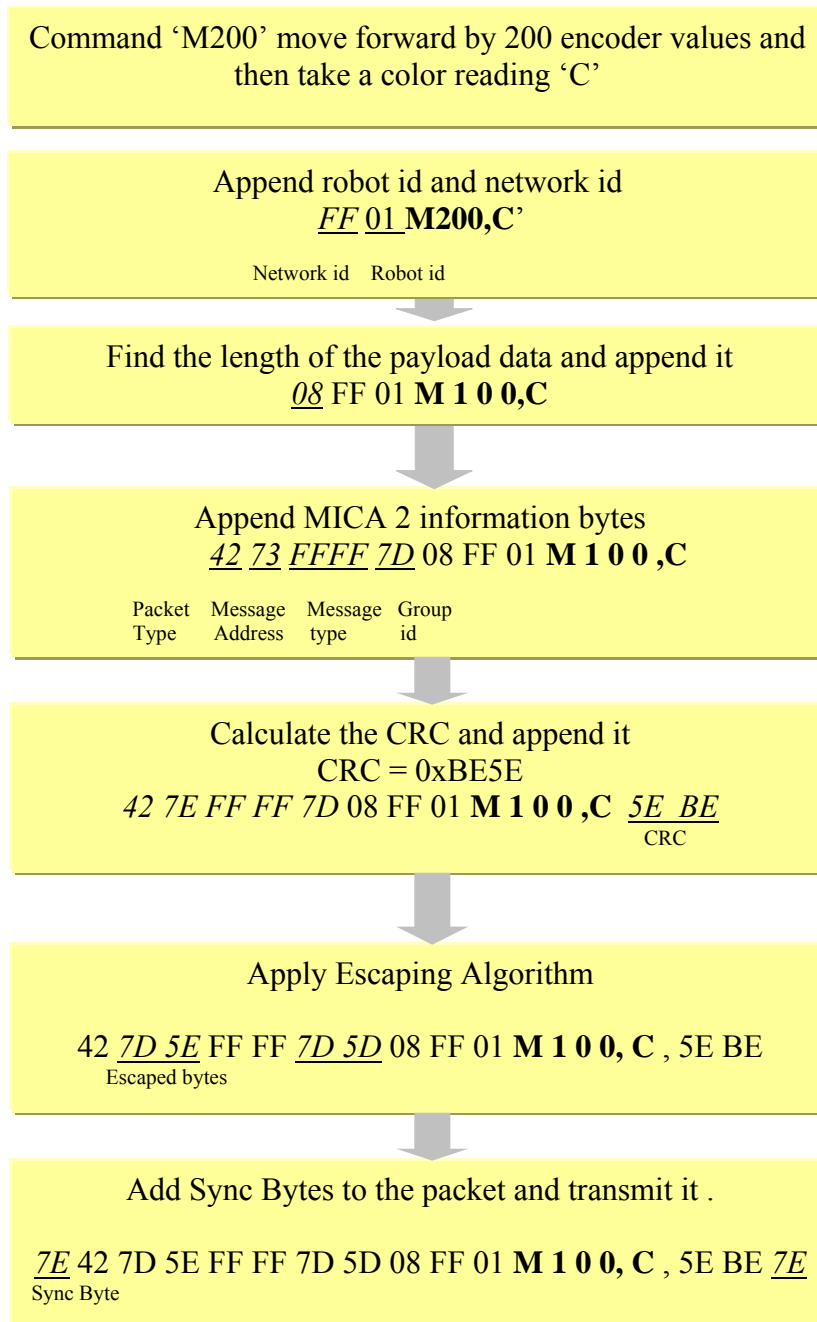


Figure 4.2 TinyOS packet formation procedure

#### 4.1.1 Challenges in wireless communication on ARRI BOT

Although Crickets offer an excellent tradeoff between cost and simplicity in structuring a WSN, they have some limitations such as the communication range and maximum payload data that can be transmitted in a packet. Some of the challenges encountered during exchanging data over a shared wireless channel are mentioned below;

##### *Limited communication range*

MICA2 and Crickets operate at a frequency of 433 MHz. The on-board antennas offer a maximum indoor coverage of 30 meters. To resolve problems related in establishing communication greater than 30 meters schemes like multi-hop networks [18] [20] have been implemented.

##### *Multi-hop Wireless Sensor Network*

To establish long distance communications with limited radio propagation range of crickets, the routes between the source and destination are generally multi-hopped to pass on data from one point to another. A general multi-hopping scheme is shown in the figure below, where there are six robots (A, B, C, D, E, F) randomly deployed with a limited radio range and are controlled by a base station.



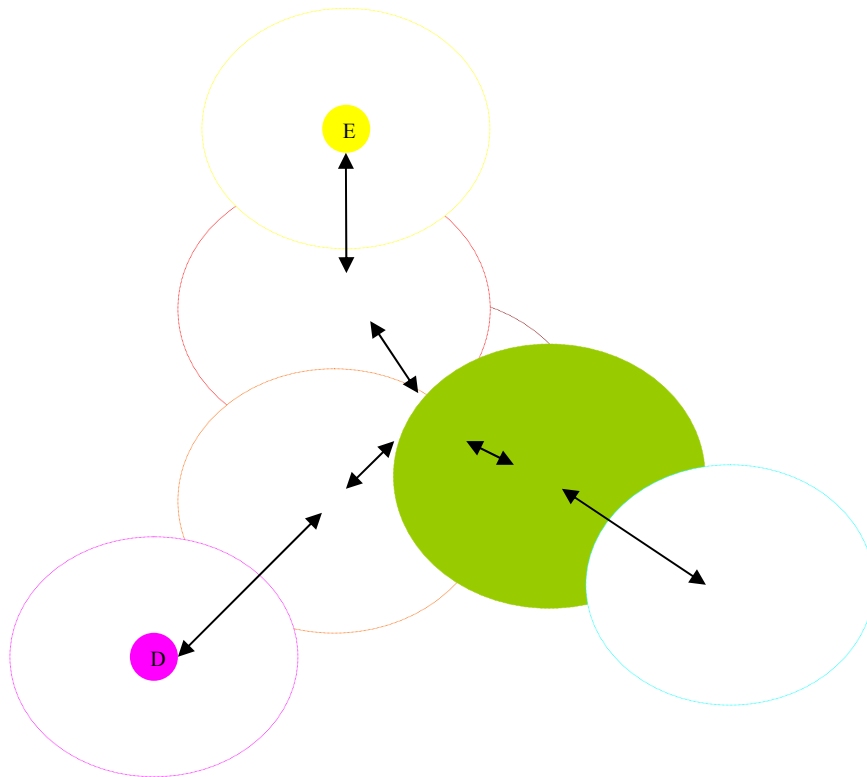


Figure 4.3 Multi-hopping with six robots

In the process of initialization the base station randomly broadcasts a wake up signal to all the robots, however only the robots A, B, C which are in range with the base station will respond to it and will form the first hopping level. The robots in Level 1 in turn transmit a wake up signal which is heard by nodes D, E and F and they form the second level in multi-hopping. In this initialization process if any mote receives more than one wake up call, then it responds to the first one only and ignores the rest. Synchronization is very important in the initialization process. This way the motes are aware of their neighboring motes and the base station knows the robots present in each hop.

With multi-hopping the power required during transmission can be minimized by reducing the communication range of the intermediate nodes.

### *Contention Control*

In case of a fully connected network topology, a common problem seen is network collision of packets. With multiple packets traveling over a common channel they are likely to collide with each other and garble the data. There are several contention control algorithms that limit the amount of transmitted traffic in a given time slot based on estimated network capacity and utilization. ALOHA [42] and Binomial Exponential back off [21] are two such congestion controls.

### *ALOHA based contention control*

Aloha protocol is a random access MAC protocol for contention control and works as follow;

- A packet is transmitted over a shared channel
- The transmitter waits for an acknowledgement from the receiver to determine if the transmission was successful or not
- The transmitter retransmits the packet after a random amount of time in case of failure in transmission.

An ALOHA based protocol has been implemented on Crickets and on the base station.

### *Limited payload*

Another challenge involved in wireless communication is the limited amount of data that can be transmitted in a TinyOS serial data packet.

With MICA2 and Crickets only 29 byte of payload data can be transmitted at a time in one packet. To get around this limitation, Huffman based compressing techniques [43] is proposed, where the amount of bits required representing a string of symbols is reduced.

### 4.2 Localizing ARRI BOT using dead reckoning

Dead reckoning on ARRI BOT is carried out by integrating the encoder value indicating the number of rotation completed by the rovers to estimate the relative position. However for this method it is imperative to have the robots precisely calibrated such that the encoder counts can be normalized to the distance traveled by the robot.

The state model for ARRI BOT is given as

$$X = [x \quad y \quad \theta]^T$$

Where  $x$  and  $y$  give the position of the robot

$\theta$  gives the orientation of the robot.

An onboard discrete time position estimator determines the position of the robot using the encoder counts  $\Delta\phi_L, \Delta\phi_R$  and the kinematic model

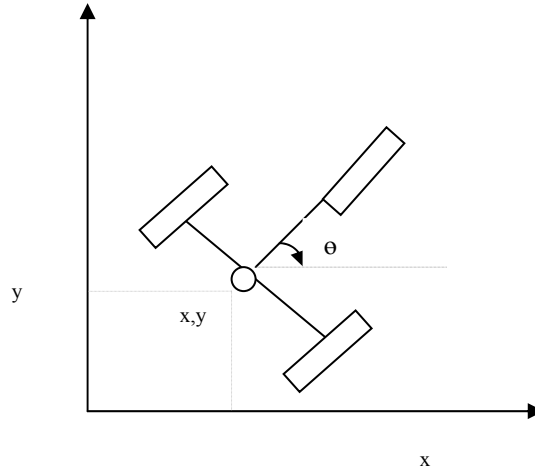


Figure 4.4 Tricycle robot configuration

$$x_{k+1} = x_k + \frac{K_{drvR} r_R \Delta\phi_R + K_{drvL} r_L \Delta\phi_L}{2} \cos(\theta_k)$$

$$y_{k+1} = y_k + \frac{K_{drvR} r_R \Delta\phi_R + K_{drvL} r_L \Delta\phi_L}{2} \sin(\theta_k)$$

$$\theta_{k+1} = \theta_k + \frac{K_{turnR} r_R \Delta\phi_R - K_{turnL} r_L \Delta\phi_L}{2L_b}$$

Where  $K_{drv}$ ,  $K_{turn}$  are the actuators drive constants to normalize the encoder values to distance traveled.

#### 4.2.1 Localization of ARRI BOT using active beacons

ARRI Bots are currently been localized using Crossbow's Crickets. They can be configured as either Beacons or Listeners. In Beacons mode they transmit a RF signal followed by a concurrent Ultrasound Pulse. The listener upon receiving the Beacons RF signal waits for the corresponding ultrasonic pulse and based on the difference in propagation speed between RF (travels at speed of light) and ultrasound it calculates the distance from the beacon. The beacons position is priory known to the listener. The listener should obtain distance information from at least two beacons to implement triangulation for estimating its position. The results from localization with Crickets are mentioned in the next chapter.

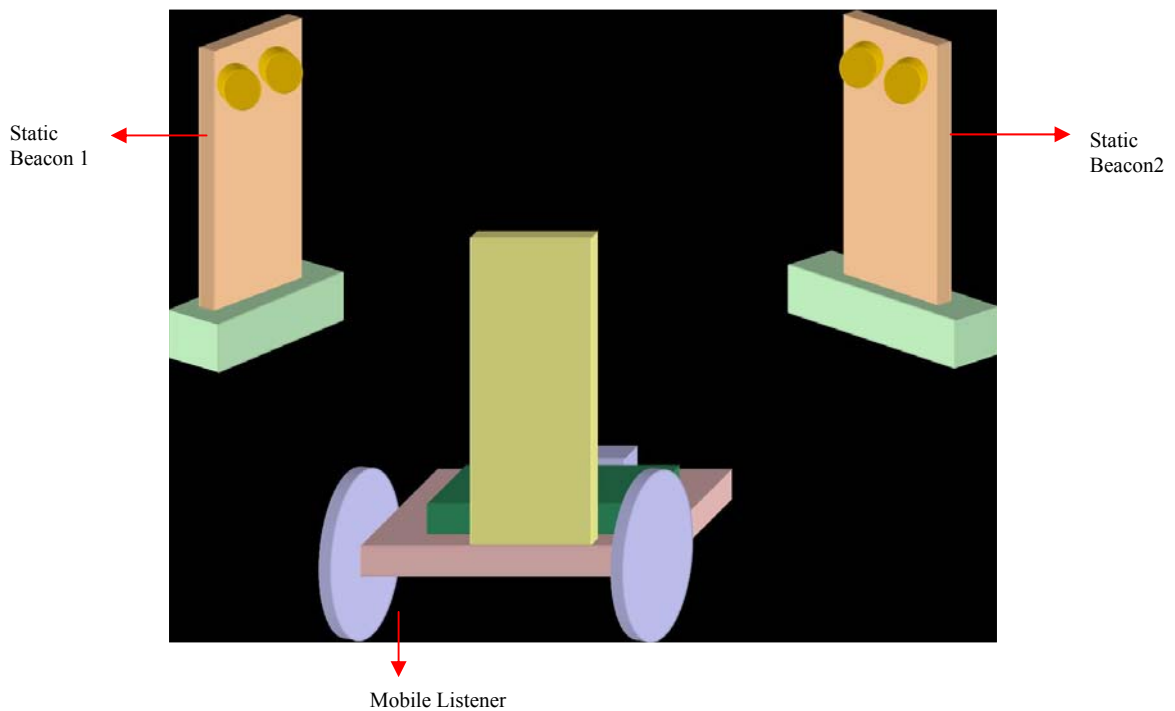


Figure 4.5 Localization of ARRI BOT using triangulation

#### 4.2.2. Challenges involved in localizing ARRI BOT

##### *Line of sight Orientation*

The ultrasonic transducers on the Crickets are not omni directional. The ultrasonic beam covers an angle of approximately  $\pm 45^\circ$  over a distance of 10.5 meters. For this reason the listeners have to be oriented in the direction of the beacons during localization. Since the servos on the robots are not precisely calibrated, this could result in orientation error for applications like trajectory following or path planning.

##### *Maintaining the Active Beacons*

Although the accuracy obtained with active beacons is much better than dead reckoning they can be expensive to maintain and consume a lot of energy to operate uninterruptedly.

#### 4.3. Sensor Fusion for localization

The GPS and Dead reckoning are two different methods to calculate location and have their own pros and cons. Dead reckoning gives the relative position estimate but the errors due to wheel slippage and uneven ground gets accumulate and become predominant after a short distance. GPS or active beacons or camera provide absolute position estimate but are expensive to maintain and are generally not Omni directional as a result of which the receivers have to maintain a line of sight distance with them. However by fusing these two methods together the weakness of each method can be counterbalanced by the other method [27].

With the resources that are currently available for ARRI BOT a fusion of dead reckoning (relative measurement) and Crossbow's Crickets (absolute measurement) is

possible using methods like the *Kalman Filter* [37] or *Bayes's Rule* and *Markov assumption* [24]

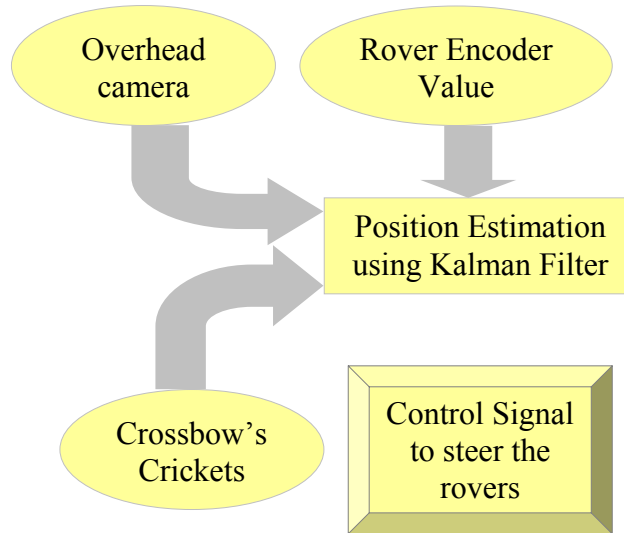


Figure 4.6 Block diagram of sensor fusion

### *Kalman Filter*

The Kalman filter is an efficient recursive filter which estimates the state of a dynamic system from a series of incomplete and noisy measurements. It has two distinct phases, the Predict phase and the Update phase. The predict phase uses the estimate from the previous time step to produce an estimate of the current state. In the update phase, measurement information from the current time step is used to correct the errors from the prediction phase.

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

$A_k$  - Is the state transition model applied on  $x_k$

$B_k$  - Is the control input model applied to  $u_k$

$w_k \sim (0, Q_k)$  is the process noise.

$H_k$  - Is the observation model which maps the true state space into the measurement space.

$$v_k \sim (0, R_k)$$

*Predict State*

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + B_k u_k$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k$$

*Update State*

$$P_{k+1} = [(P_{k+1}^-)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}]^{-1}$$

$$\hat{x}_{k+1} = \hat{x}_{k+1}^- + P_{k+1} H_{k+1}^T R_{k+1}^{-1} (z_{k+1} - H_{k+1} \hat{x}_{k+1}^-)$$



#### 4.4 Adaptive Sampling using ARRI BOT

A linear stationary parametric field is printed (142 inches x 106 inches) whose coefficients are to be estimated using the color sensor on ARRI Bot and Kalman filter.

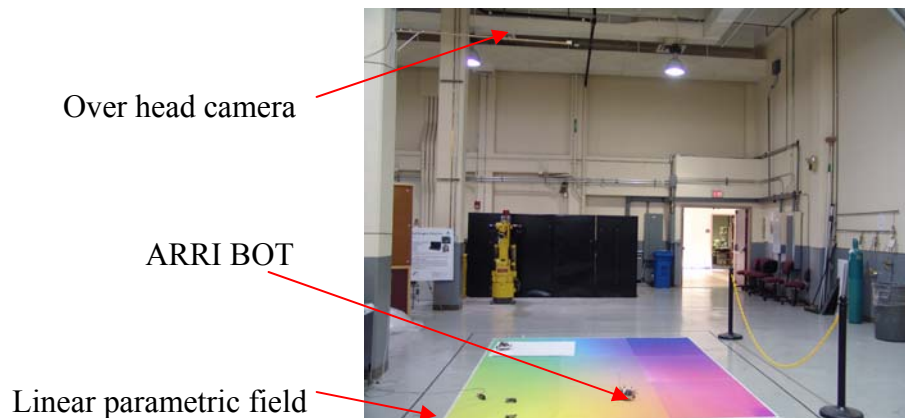


Figure 4.7 Linear parametric field for validating Adaptive sampling

An overhead camera placed at a height of 13 feet, is used to capture the entire color field in its view. A base station is used to serve as a central processing unit for executing sampling algorithms and to maintain communication with the robots.

The color sensor on the ARRI BOT acquires color reading and transmits it to the base station. A MATLAB® based program, uses the sensor information to calculate error covariance using Kalman filter. It then decides the next sampling location, to achieve minimum error covariance. The next sampling location is transmitted to the robot, which then carries out the maneuverability required to reach the destination. The overhead camera, which is also interfaced with the base station, provides information on the exact location of robot.

The MATLAB® code, running on the base station, is responsible for receiving and transmitting packet to the robots. The code also processes the image obtained from the overhead camera and simultaneously executes the Kalman filter based adaptive sampling algorithm. The Javelin stamp on the robots is responsible for executing commands obtained from the base station and also computing the dead reckoning to estimate its position.

Assuming the field distribution to be linear in its parameters, a closed form solution is achieved, to decide the next sampling location based on previous measurements.

The field parameters can be described as,

$$R = r_0 + r_1x + r_2y$$

$$G = g_0 + g_1x + g_2y$$

$$B = b_0 + b_1x + b_2y$$

where, R, G, B stands for Red, Green and Blue respectively.

$r_0, r_1, r_2, g_0, g_1, g_2, b_0, b_1, b_2$  are the unknown field parameters

x, y are the known positions at which the robot samples.

The proportion of R, G, and B at a sampled point varies from 0 to 1. The color sensor provides a value from 0 to 255 which is normalized to 0 to 1 to compute the field parameters.

Extended Kalman filter is used to determine the next sampling position based on the error covariance from the previous measurement update.

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

Where,  $A_k$  - Is the state transition model applied on  $x_k$

$B_k$  - Is the control input model applied to  $u_k$

$w_k \sim (0, Q_k)$  is the process noise.

$H_k$  - Is the observation model which maps the true state space into the measurement space.

$$v_k \sim (0, R_k)$$

*Predict State*

$$\hat{x}_{k+1}^- = A_k \hat{x}_k + B_k u_k$$

$$P_{k+1}^- = A_k P_k A_k^T + Q_k$$

*Update State*

$$P_{k+1} = [(P_{k+1}^-)^{-1} + H_{k+1}^T R_{k+1}^{-1} H_{k+1}]^{-1}$$

$$\hat{x}_{k+1} = \hat{x}_{k+1}^- + P_{k+1} H_{k+1}^T R_{k+1}^{-1} (z_{k+1} - H_{k+1} \hat{x}_{k+1}^-)$$

## SYSTEM MODEL

$$\begin{pmatrix} X_{k+1} \\ A_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + \begin{pmatrix} B_k \\ 0 \end{pmatrix} U_k + \begin{pmatrix} w_k \\ 0 \end{pmatrix}$$

$$\begin{array}{l} \text{Robots} \\ \text{Position and} \\ \text{Orientation} \\ \text{Field} \\ \text{Parameters} \end{array} \begin{array}{l} \left\{ \begin{array}{l} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{array} \right\}_{k+1} \\ \left\{ \begin{array}{l} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{array} \right\}_k \end{array} = \begin{array}{l} \left[ \begin{array}{cc} \frac{K_R r_R}{2} \cos(\theta_k) & \frac{K_L r_L}{2} \cos(\theta_k) \\ \frac{K_R r_R}{2} \sin(\theta_k) & \frac{K_L r_L}{2} \sin(\theta_k) \\ \frac{K_R r_R}{2L_b} & -\frac{K_L r_L}{2L_b} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \begin{array}{l} \left[ \begin{array}{l} \Delta\phi_R \\ \Delta\phi_L \end{array} \right]_k + \left[ \begin{array}{l} w_k \\ 0 \end{array} \right] \end{array}$$

## TIME UPDATE

The time update of the robots position based on dead reckoning is given as

$$\begin{array}{l} \hat{x} \\ y \\ \hat{\theta} \\ \hat{r}_0 \\ \hat{r}_1 \\ \hat{r}_2 \\ \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{array} \Big|_{k+1} = \begin{array}{l} \hat{x} \\ \hat{y} \\ \hat{\theta} \\ \hat{r}_0 \\ \hat{r}_1 \\ \hat{r}_2 \\ \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{array} \Big|_k + \begin{array}{l} \left[ \begin{array}{cc} \frac{K_R r_R}{2} \cos(\hat{\theta}_k) & \frac{K_L r_L}{2} \cos(\hat{\theta}_k) \\ \frac{K_R r_R}{2} \sin(\hat{\theta}_k) & \frac{K_L r_L}{2} \sin(\hat{\theta}_k) \\ \frac{K_R r_R}{2L_b} & -\frac{K_L r_L}{2L_b} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array} \right] \begin{array}{l} \left[ \begin{array}{l} \Delta\phi_R \\ \Delta\phi_L \end{array} \right]_k \end{array}$$

$$\begin{aligned}
A &= I_{12} \\
P_{k+1}^- &= P_k + Q_k \\
\hat{X}_{k+1}^- &= \hat{X}_k + \hat{B}_k U_k \\
\hat{A}_{k+1}^- &= \hat{A}_k
\end{aligned}$$

### MEASUREMENT MODEL

$$\begin{aligned}
\tilde{z}_k &= h_k(X_k, A_k) + v_k \\
\begin{bmatrix} \tilde{r}_k \\ \tilde{g}_k \\ \tilde{b}_k \end{bmatrix} &= \begin{bmatrix} r_{0k} + r_{1k}x_k + r_{2k}y_k \\ g_{0k} + g_{1k}x_k + g_{2k}y_k \\ b_{0k} + b_{1k}x_k + b_{2k}y_k \end{bmatrix} + v_k
\end{aligned}$$

### MEASUREMENT UPDATE

The measurements from the color sensor and the over head camera (position) are used to update the model and find the error covariance which will decide the next best sampling location.

$$\begin{aligned}
\hat{H}_{k+1} &= \frac{\partial \hat{h}_{k+1}}{\partial (\hat{X}_{k+1}^-, \hat{A}_{k+1}^-)} = \begin{bmatrix} \hat{r}_{1k+1}^- & \hat{r}_{2k+1}^- & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{g}_{1k+1}^- & \hat{g}_{2k+1}^- & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 \\ \hat{b}_{1k+1}^- & \hat{b}_{2k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- \end{bmatrix} \\
K_{k+1} &= P_{k+1}^- \hat{H}_{k+1}^T (\hat{H}_{k+1} P_{k+1}^- \hat{H}_{k+1}^T + R_{k+1})^{-1} \\
P_{k+1} &= (I_{12} - K_{k+1} \hat{H}_{k+1}) P_{k+1}^- \\
\begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} &= \begin{pmatrix} \hat{X}_{k+1}^- \\ \hat{A}_{k+1}^- \end{pmatrix} + K_{k+1} \{ \tilde{z}_{k+1} - \hat{h}_{k+1} \} = \begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} + K_{k+1} \left\{ \begin{bmatrix} \tilde{r}_{k+1} \\ \tilde{g}_{k+1} \\ \tilde{b}_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{r}_{0k+1}^- + \hat{r}_{1k+1}^- \hat{x}_{k+1}^- + \hat{r}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{g}_{0k+1}^- + \hat{g}_{1k+1}^- \hat{x}_{k+1}^- + \hat{g}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{b}_{0k+1}^- + \hat{b}_{1k+1}^- \hat{x}_{k+1}^- + \hat{b}_{2k+1}^- \hat{y}_{k+1}^- \end{bmatrix} \right\}
\end{aligned}$$

The results of adaptive sampling are mentioned in the next chapter.

### *Adaptive Sampling with multiple robots*

In order to achieve faster coverage of the test bed with minimum number of samples a multi-robot adaptive sampling [36] is implemented. Some of the advantages with multiple robot sampling are

- The total time required to complete the mission is reduced
- Energy consumption per robot is optimized
- Increase in the coverage and connectivity
- The robots can localize themselves efficiently by information sharing
- With increase in the available information the susceptibility to errors

is reduced thus providing a detail information of the unknown parameters

The multi-robot adaptive sampling algorithm is tested based on two case studies

1. Considering no localization uncertainty
2. Considering Localization uncertainty
  - a. Considering Camera Measurement Uncertainty
  - b. Considering only Color sensor measurement to estimate the field parameters and localization (no camera).

#### *1. Considering no localization uncertainty*

In this method, only the parameters of the linear field is estimated using Kalman filter where as the position is measured using the overhead camera assuming it to be the true location of the robot. Dead reckoning is not used in this case.

The system model is now

$$A_{k+1} = A_k$$

The effect of system model is given as

$$\begin{aligned}\hat{A}_{k+1}^- &= \hat{A}_k \\ P_{k+1}^- &= P_k\end{aligned}$$

*Measurement model*

$$\begin{aligned}\tilde{z}_k &= h_k(A_k) + v_k \\ \begin{bmatrix} \tilde{r}_k \\ \tilde{g}_k \\ \tilde{b}_k \end{bmatrix} &= \begin{bmatrix} r_{0k} + r_{1k}x + r_{2k}y \\ g_{0k} + g_{1k}x + g_{2k}y \\ b_{0k} + b_{1k}x + b_{2k}y \end{bmatrix} + v_k\end{aligned}$$

*Measurement Update*

$$H_{k+1} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & y \end{bmatrix}$$

$$K_{k+1} = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1})^{-1}$$

$$P_{k+1} = (I_{12} - K_{k+1} H_{k+1}) P_{k+1}^-$$

$$\hat{A}_{k+1} = \hat{A}_{k+1}^- + K_{k+1} \left\{ \tilde{z}_{k+1} - \hat{h}_{k+1} \right\} = \hat{A}_{k+1}^- + K_{k+1} \left\{ \begin{bmatrix} \tilde{r}_{k+1} \\ \tilde{g}_{k+1} \\ \tilde{b}_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{r}_{0k+1}^- + \hat{r}_{1k+1}^- x + \hat{r}_{2k+1}^- y \\ \hat{g}_{0k+1}^- + \hat{g}_{1k+1}^- x + \hat{g}_{2k+1}^- y \\ \hat{b}_{0k+1}^- + \hat{b}_{1k+1}^- x + \hat{b}_{2k+1}^- y \end{bmatrix} \right\}$$

*2a. Considering localization uncertainty in the camera measurement*

In this case the field parameters and the robots position are estimated using EKF. The position measurement is taken from the overhead camera considering some embedded error. Dead reckoning is used during the time update phase of the EKF to estimate the next sampling location.

The system model is now

$$\begin{pmatrix} X_{k+1} \\ A_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + \begin{pmatrix} B_k \\ 0 \end{pmatrix} U_k + \begin{pmatrix} w_k \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}_k + \begin{bmatrix} \frac{K_R r_R}{2} \cos(\theta_k) & \frac{K_L r_L}{2} \cos(\theta_k) \\ \frac{K_R r_R}{2} \sin(\theta_k) & \frac{K_L r_L}{2} \sin(\theta_k) \\ \frac{K_R r_R}{2L_b} & -\frac{K_L r_L}{2L_b} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\phi_R \\ \Delta\phi_L \end{bmatrix}_k + \begin{bmatrix} w_k \\ 0 \end{bmatrix}$$

*Effect of system model*

$$w_k \sim N(0, Q), \quad \xi_k \sim N(0, R_1), \quad v_k \sim N(0, R_2), \quad x_0 \sim N(\bar{x}_0, P_0)$$



$$A = I_{12}$$

$$P_{k+1}^- = P_k + Q_k$$

$$\hat{X}_{k+1}^- = \hat{X}_k + \hat{B}_k U_k$$

$$\hat{A}_{k+1}^- = \hat{A}_k$$

$$\begin{bmatrix} \hat{x} \\ y \\ \hat{\theta} \\ \hat{r}_0 \\ \hat{r}_1 \\ \hat{r}_2 \\ \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{bmatrix}_{k+1} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \\ \hat{r}_0 \\ \hat{r}_1 \\ \hat{r}_2 \\ \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{bmatrix}_k + \begin{bmatrix} \frac{K_R r_R}{2} \cos(\hat{\theta}_k) & \frac{K_L r_L}{2} \cos(\hat{\theta}_k) \\ \frac{K_R r_R}{2} \sin(\hat{\theta}_k) & \frac{K_L r_L}{2} \sin(\hat{\theta}_k) \\ \frac{K_R r_R}{2L_b} & -\frac{K_L r_L}{2L_b} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\phi_R \\ \Delta\phi_L \end{bmatrix}_k$$

The measurement model is now

$$\begin{bmatrix} \tilde{Y}_k \\ \tilde{Z}_k \end{bmatrix} = \begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \\ \tilde{\theta}_k \\ \tilde{r}_k \\ \tilde{g}_k \\ \tilde{b}_k \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \\ r_{0k} + r_{1k}x_k + r_{2k}y_k \\ g_{0k} + g_{1k}x_k + g_{2k}y_k \\ b_{0k} + b_{1k}x_k + b_{2k}y_k \end{bmatrix} + \begin{pmatrix} \xi_k \\ v_k \end{pmatrix}$$

*Measurement update*

$$\hat{H}_{k+1} = \frac{\partial \hat{h}_{k+1}}{\partial (\hat{X}_{k+1}^-, \hat{A}_{k+1}^-)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{r}_{1k+1}^- & \hat{r}_{2k+1}^- & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{g}_{1k+1}^- & \hat{g}_{2k+1}^- & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 \\ \hat{b}_{1k+1}^- & \hat{b}_{2k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- \end{bmatrix}$$

$$K_{k+1} = P_{k+1}^- \hat{H}_{k+1}^T (\hat{H}_{k+1} P_{k+1}^- \hat{H}_{k+1}^T + R_{k+1})^{-1}$$

$$P_{k+1} = (I_{12} - K_{k+1} \hat{H}_{k+1}) P_{k+1}^-$$

$$\begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{X}_{k+1}^- \\ \hat{A}_{k+1}^- \end{pmatrix} + K_{k+1} \left\{ \begin{bmatrix} \tilde{Y}_{k+1} \\ \tilde{Z}_{k+1} \end{bmatrix} - \hat{h}_{k+1} \right\} = \begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} + K_{k+1} \left\{ \begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{y}_{k+1} \\ \tilde{\theta}_{k+1} \\ \tilde{r}_{k+1} \\ \tilde{g}_{k+1} \\ \tilde{b}_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{x}_{k+1}^- \\ \hat{y}_{k+1}^- \\ \hat{\theta}_{k+1}^- \\ \hat{r}_{0k+1}^- + \hat{r}_{1k+1}^- \hat{x}_{k+1}^- + \hat{r}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{g}_{0k+1}^- + \hat{g}_{1k+1}^- \hat{x}_{k+1}^- + \hat{g}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{b}_{0k+1}^- + \hat{b}_{1k+1}^- \hat{x}_{k+1}^- + \hat{b}_{2k+1}^- \hat{y}_{k+1}^- \end{bmatrix} \right\}$$

2b. *Considering only Color sensor measurement to estimate the field parameters and localization (no camera).*

In this case only the color sensor measurements are used to estimate the field parameters and localization using EKF. The camera measurement is used to compare the true and estimated position. Dead reckoning is running in background for time update estimation.

The system model is given as

$$\begin{pmatrix} X_{k+1} \\ A_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + \begin{pmatrix} B_k \\ 0 \end{pmatrix} U_k + \begin{pmatrix} w_k \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}_k + \begin{bmatrix} \frac{K_R r_R}{2} \cos(\theta_k) & \frac{K_L r_L}{2} \cos(\theta_k) \\ \frac{K_R r_R}{2} \sin(\theta_k) & \frac{K_L r_L}{2} \sin(\theta_k) \\ \frac{K_R r_R}{2L_b} & -\frac{K_L r_L}{2L_b} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\phi_R \\ \Delta\phi_L \end{bmatrix}_k + \begin{bmatrix} w_k \\ 0 \end{bmatrix}$$

*Measurement update*

$$\hat{H}_{k+1} = \frac{\partial \hat{h}_{k+1}}{\partial (\hat{X}_{k+1}^-, \hat{A}_{k+1}^-)} = \begin{bmatrix} \hat{r}_{1k+1}^- & \hat{r}_{2k+1}^- & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{g}_{1k+1}^- & \hat{g}_{2k+1}^- & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 \\ \hat{b}_{1k+1}^- & \hat{b}_{2k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- \end{bmatrix}$$

$$K_{k+1} = P_{k+1}^- \hat{H}_{k+1}^T (\hat{H}_{k+1} P_{k+1}^- \hat{H}_{k+1}^T + R_{k+1})^{-1}$$

$$P_{k+1} = (I_{12} - K_{k+1} \hat{H}_{k+1}) P_{k+1}^-$$

$$\begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{X}_{k+1}^- \\ \hat{A}_{k+1}^- \end{pmatrix} + K_{k+1} \{ \tilde{z}_{k+1} - \hat{h}_{k+1} \} = \begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} + K_{k+1} \left\{ \begin{bmatrix} \tilde{r}_{k+1} \\ \tilde{g}_{k+1} \\ \tilde{b}_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{r}_{0k+1}^- + \hat{r}_{1k+1}^- \hat{x}_{k+1}^- + \hat{r}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{g}_{0k+1}^- + \hat{g}_{1k+1}^- \hat{x}_{k+1}^- + \hat{g}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{b}_{0k+1}^- + \hat{b}_{1k+1}^- \hat{x}_{k+1}^- + \hat{b}_{2k+1}^- \hat{y}_{k+1}^- \end{bmatrix} \right\}$$

Raster scanning and Greedy Adaptive sampling was also tested using ARRI

BOT, the comparison of these three methods are mentioned in the next chapter.

## CHAPTER 5

### EXPERIMENTAL RESULTS

The previous chapters discussed the features and deployment algorithm implemented on ARRI BOT. In this chapter the results obtained from the deployment algorithms like localization and adaptive sampling along with the results obtained from the energy harvesting module on ARRI BOT are presented.

#### 5.1 ARRIBOT in Localization

Extensive and repeated testing has been carried out to observe the performance of localization algorithm using ARRI BOTs and Crossbow's Crickets. A Passive Mobile Architecture is used to check the accuracy and analyze the uncertainties involved in localization. A JAVA based interface is developed to establish communication with the robot using a Cricket base station. The interface records the distance information from the robots and computes its x y coordinates using triangulation. It also compares the measured values with the actual value obtained either from overhead camera or using a measuring tape.

##### *Case I – Localizing a single robot with three static beacons*

In this experiment, the listener on the BOT, 'L' obtains distance information from three static beacons, B1, B2, and B3 whose positions are priory known. The mobile listener takes the distance measurements from the three beacons and transmits it to the base station. In order to get around the line of sight contact constraint the robot is

made to rotate till it gets the distance value from all the three beacons and then moves to a different position.

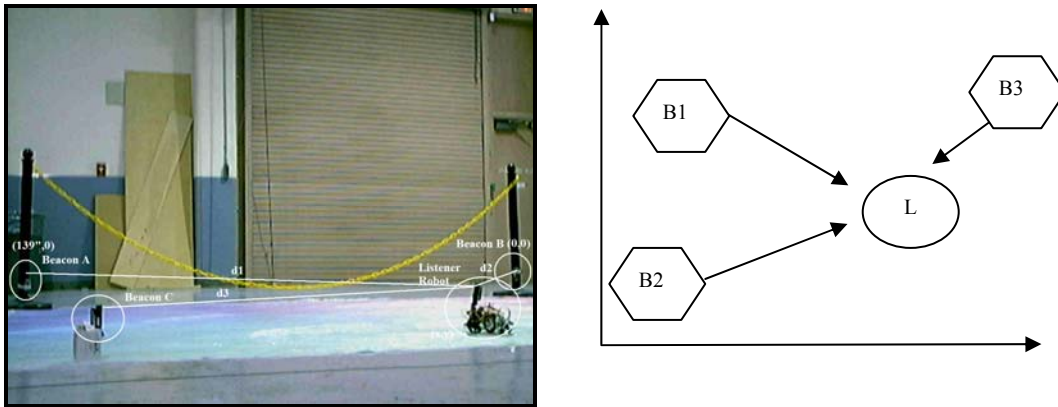


Figure 5.1 Experimental setup for testing localization accuracy

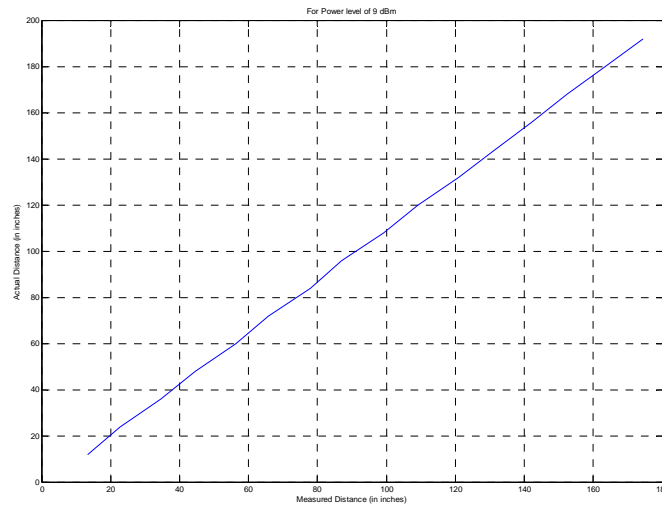


Figure 5.2 Actual versus measured distance obtained from the crickets

Table 5.1 Comparison of the actual distance and distance from the crickets

<b>Actual Distance (in inches)</b>	<b>From Cricket (in inches)</b>
12	13.3858
24	22.8346
36	34.6457
48	44.4882
60	56.2992
72	65.748
84	77.9528
96	87.0079
108	99.2126
120	109.0551
132	120.8661
144	131.4961
156	142.126
168	152.3622
180	163.3858
192	174.4094

From the table above, it was observed that as the distance from the beacons increased, the error in the measured value obtained from the listener also increased.

Error = Actual value (measured using overhead camera)-Measured value (from mote)

In order to offset this constant error a correction factor was added to the Crickets distance calculation program.

$$d_{actual} = 1.1141.d_{measured} - 2.1150 \longrightarrow \text{Correction factor}$$

*Case II – Localization of a single robot using the correction factor and two static beacons*

The same experiment mentioned in case I was repeated using the correction factor derived from the previous experiment. This time the listener L on the robot localizes itself from only two static beacons B1 and B2 whose positions are priority know. The actual and measured x y coordinates obtained are mentioned in Table

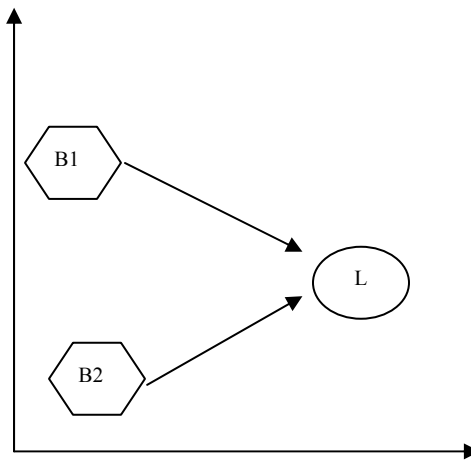


Figure 5.3 Experimental Setup for testing localization using triangulation

The figure below shows the comparison of the actual and measured position of the BOT taken at different locations

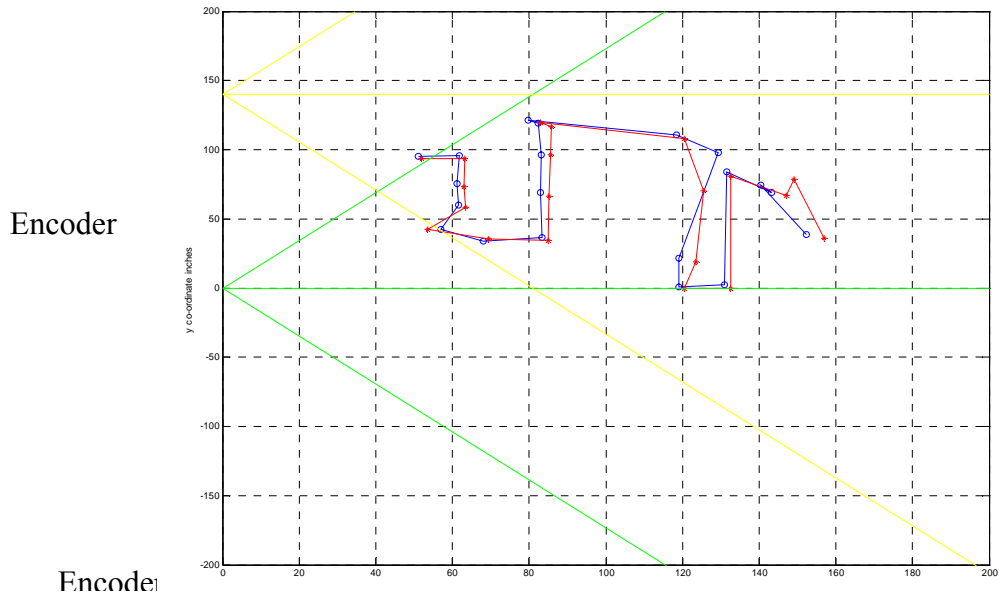


Figure 5.4 Plot of actual and measured position with correction factor

The ‘\*’ are the actual position of ARRI BOT where as the ‘o’ are the measured position recorded from the crickets. The green and the yellow lines represent the ultrasonic beam coverage of B1 and B2 respectively. The error in the actual and measured x, y coordinates is shown below.

*Case III– Localization over time using two static Beacons and three stationary ARRIBOTs*

This experiment was conducted to observe any occurrence of error due to variation in time. In order to achieve line of sight constraint the robots were placed facing the beacons instead of rotating them.



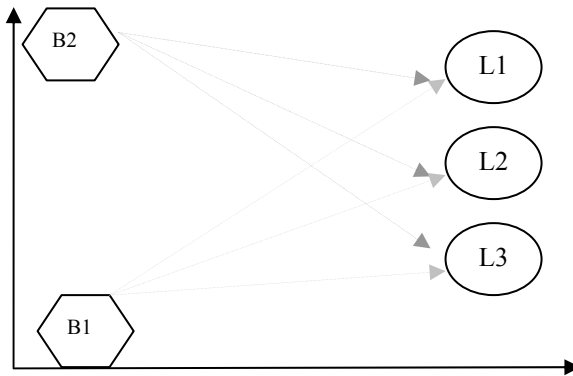


Figure 5.5 Experimental Setup with three robots

This experiment was conducted to observe any occurrence of error due to variation in time. In order to achieve line of sight constraint the robots were placed facing the beacons instead of rotating them.

The position recorded at different time instance for three robots is shown in the figure below.

It was been observed that the distance values obtained from the listeners were consistent with a small variation in the range of 1cm to 2cm over time.

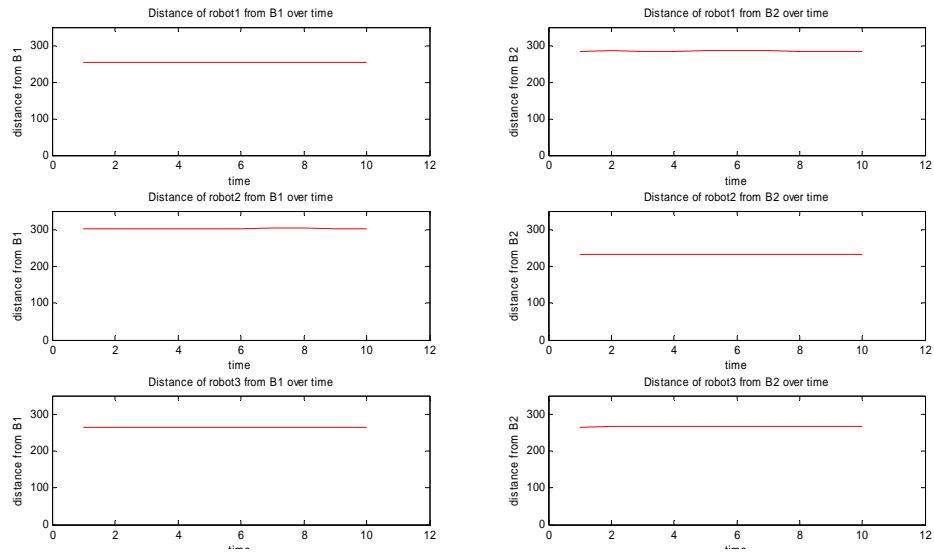


Figure 5.6 x y position recorded from three robots at different time instant

### *ARRI BOT Localization demo*

The objective of ARRI BOT localization demo was to have the robots navigate in a formation that would help them to localize with each other and using minimum number of static beacons.

### *Constraints*

1. The robot to be localized needs distance information from at least two beacons whose positions are priory know in order to carry out the triangulation process.
2. The distance between the robots should be such that there is line of sight contact maintained at the time of localization.

3. The initial positions of at least two static/mobile beacons are known.
4. The robots have to be in range with the base station so that data and command packets can be transmitted and received.

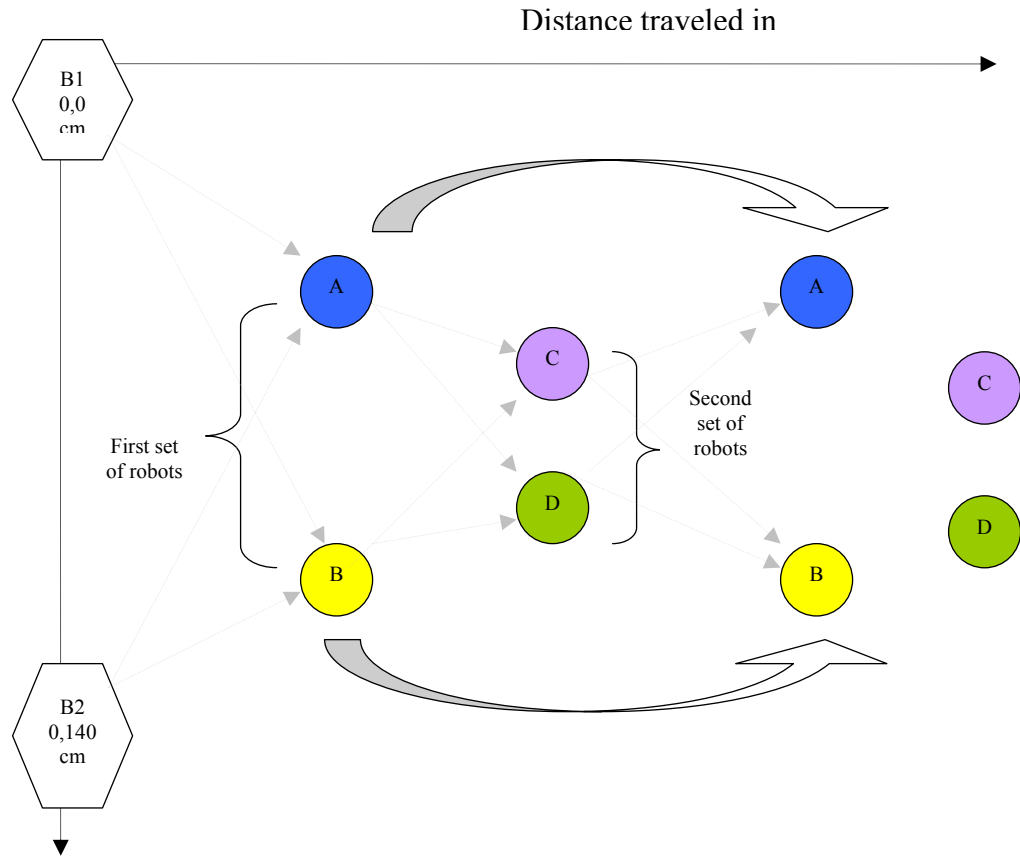


Figure 5.7 Experimental Setup for localization demo

### *The approach*

➤ Four robots and two static beacons are used for this demo. The robot to be localized obtains distance information from at least two beacons and reports it to the base station.

➤ All the controls and computations are centralized at the base station for this approach. However, in future the crickets and javelin stamp on each robot could contribute in computing the position on-board and sharing information among the robots.

➤ It is assumed that the robots are unaware of their initial position, however the position of the two static beacons is known at the start.

➤ The four robots are divided into two sets, with two robots in each set. The first set consists of robots A and B and the second set consists of robots C and D.

➤ At a given instance only one set of robots navigates while the other set is stationary.

➤ In the beginning the first set of robots localizes itself with respect to the base station, meanwhile the second set navigates within the quadrant maintaining line of sight contact with the first set of robots.

➤ Once the second set of robots travel a certain distance, the stationary or the first set of robots whose positions are now known with respect to the static beacons will switch into beacon mode to help the second set to localize.

➤ Once the first set finish localizing the second set of robots, they will cross pass the second set and navigate in a new area. The second set of robots will now become the stationary set.

➤ Once the first set of robots travels a certain distance, the stationary robot switches into beacon mode to localize robots A and B.

➤ This process is repeated until a specified area is navigated or if the robots go out of range with respect to the base station.

### 5.2 ARRI BOT in energy harvesting

ARRI BOT was repeatedly tested for its energy harvesting efficiency under varying environment conditions.

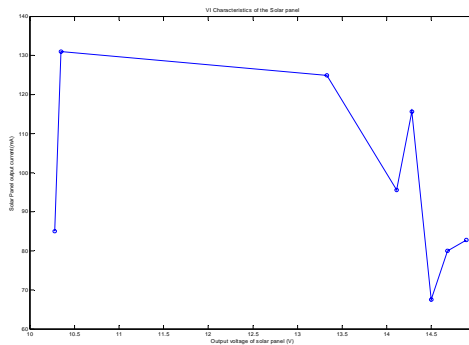


Figure 5.8 The VI characteristic of the solar panel

It was been observed that the output of the solar panel varies with variation in the intensity of the sunlight. This in turn varies the amount of time taken to charge the batteries.

Table 5.2 Time taken to charge by the batteries

<b>Battery type</b>	<b>Time taken to charge using solar panel</b>
9v , 150mAh NiMH	45 - 60 min
6V, 4000mAh NiMH	7 – 8 hours

With the technology currently available in manufacturing solar cells only 11% to 14% of the incident sunlight can be transformed into electrical energy. Which means that obtain high output the size of the solar panel should be bigger. However, the increase in size will reduce the charging time but the portability property will also be reduced which is not favorable. Hence a tradeoff has to be made between the size or the output capacity of the panel and the time required to charge a battery.

Piezoelectric transducers were tested to capture the vibration created by the robot to charge the batteries. A panel with multiple piezoelectric transducers was designed to capture vibration along the x, y, z direction of the ARRI BOT. The output measured was very low in the order of 0.01mWatts.

### 5.3 ARRI BOT in Adaptive sampling

EKF based sampling method like the Raster scanning, Adaptive sampling and Greedy Adaptive sampling are verified using ARRI BOTS. Simulation and experimental results are used to compare these sampling methods.

#### *Experimental setup*

The experimental setup required to validate the sampling methods mainly consists of a linear parametric color fields whose parameter are to be estimated, a ARRI BOT with color sensor to provide information on the unknown parameters, optical

encoder for dead reckoning and a overhead wide-angle camera to determine the true position of the robot.

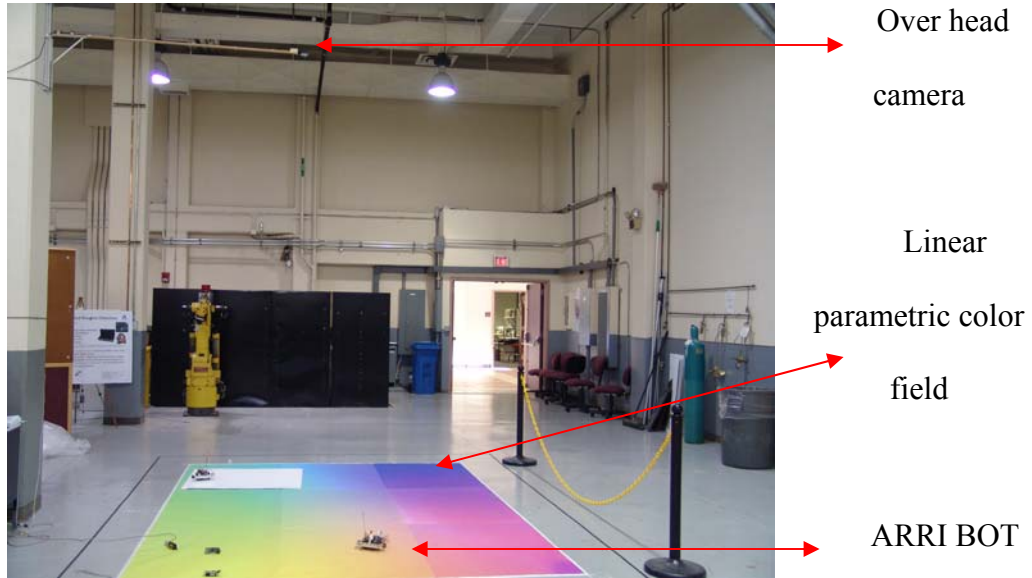


Figure 5.9 Experimental setup for testing adaptive sampling

The linear parametric color field is generated using the primary RGB colors,

$$R = r_0 + r_1x + r_2y$$

$$G = g_0 + g_1x + g_2y$$

$$B = b_0 + b_1x + b_2y$$

Where, R, G, B stands for Red, Green and Blue respectively.

$$r_0 = 0.2307, r_1 = 0.0012, r_2 = -0.00048, g_0 = 0, g_1 = 0.0002, g_2 = 0.0018,$$

$$b_0 = 1.0, b_1 = -0.00078, b_2 = -0.001.$$

### Simulation Results

Simulation results are used to compare the performance of raster scanning, adaptive sampling and greedy adaptive sampling with respect to number of samples required and the time take to estimate the unknown parameters.

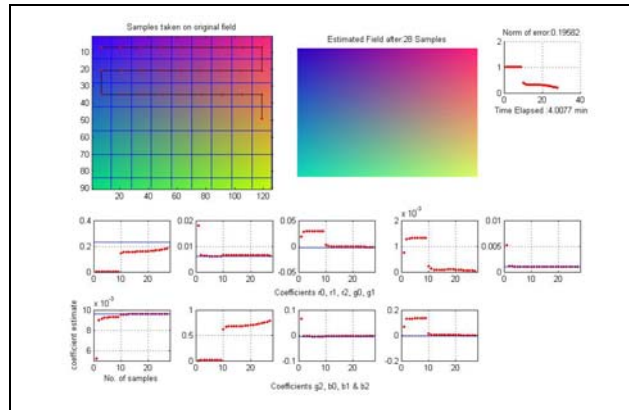


Figure 5.10 Simulation results from raster scanning

With raster scanning, 28 samples are required to recreate the linear parametric field.

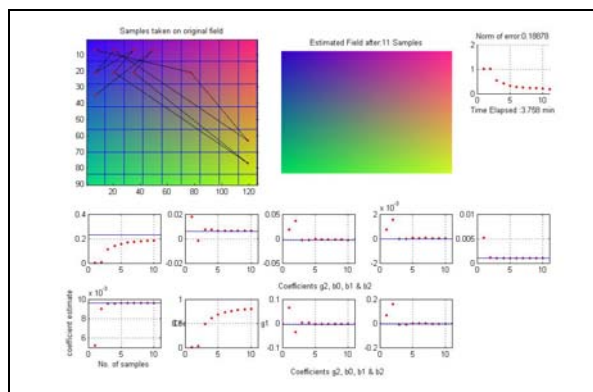


Figure 5.11 Simulation results from adaptive sampling.

With adaptive sampling, 11 samples were required to recreate the linear parametric field.



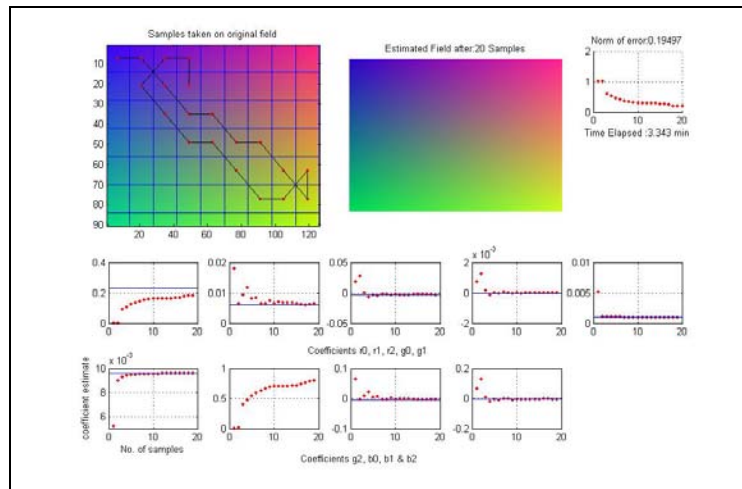


Figure 5.12 Simulation results from greedy adaptive sampling.

With greed adaptive sampling method, 20 samples were required to regenerate the linear parametric field.

Table 5.3 Simulation comparison of different sampling methods

Method	Number of samples	Estimation time
Raster scan	28	4.0077 min
Adaptive sampling	11	3.758 min
Greedy AS	20	3.343 min

### *Experimental results*

For experimental results, a base station is used to send commands to move the robot to a location estimated by the sampling algorithm running in MATLAB®. The robot navigates to the estimated position using dead reckoning. Measurements from the camera are used to correct the dead reckoning errors in the update stage of EKF.

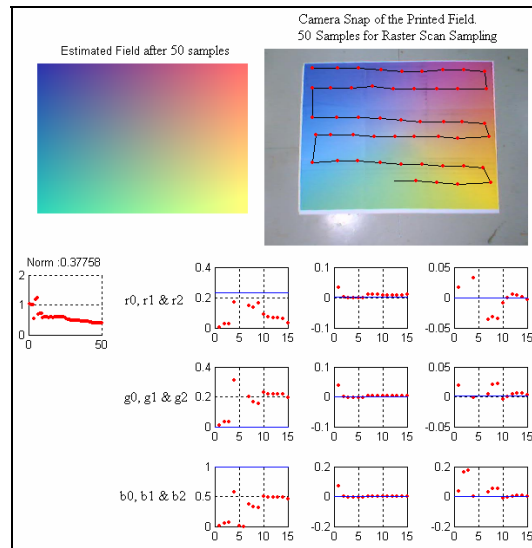


Figure 5.13 Experimental results from raster scanning

With raster scanning, 50 samples are required to recreate the linear parametric field.

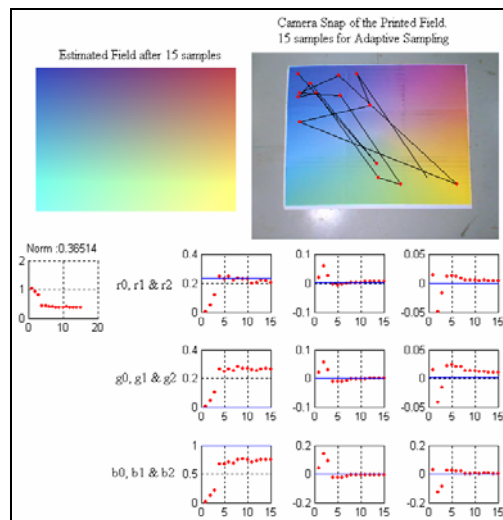


Figure 5.14 Experimental results from adaptive sampling (AS)

With adaptive sampling, 15 samples were required to recreate the linear parametric field.

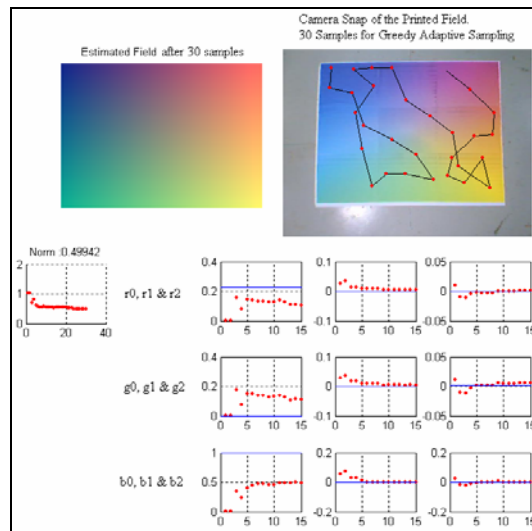


Figure 5.15 Experimental results from greedy adaptive sampling (GAS)

With greedy adaptive sampling method, 30 samples were required to regenerate the linear parametric field.

Table 5.4 Experimental comparison of sampling algorithms

Method	Number of samples (n)	2-norm of error in coefficients after n samples
Raster scan	50	0.378
Adaptive sampling	15	0.365
Greedy AS	30	0.499

Experimental and simulation results show that the greedy AS algorithm recovers the color field in minimum time with optimal number of samples.

## CHAPTER 6

### CONCLUSION

This chapter summarizes the contribution of this thesis and also provides suggestions for future work in the field of designing a mobile wireless sensor node.

#### 6.1 Thesis contribution

This thesis has helped in catering the demands for a mobile wireless sensor node. With the new upgrade incorporated, the indigenously developed ARRI BOT evolves into a high tech mobile sensor node which can be easily deployed in a wireless network along with other robotic platforms.

This thesis has helped in developing a reliable and robust wireless protocol using Crossbows® motes and schemes like retransmission based on acknowledgement, sequence numbers to keep a record of the amount of packets transmitted and received, broadcast and unicast addressing and error detection based on CRC computation.

The thesis has also contributed in providing an interface for two ultrasonic ranging modules for navigational purposes. The modules provide precise information on distance within a range of 3cm to 3m and  $\pm 20^\circ$  angle wide.

The thesis has also explored two different methods of energy harvesting to charge the on-board batteries. First method used solar cells on the BOT to charge the 6V, 4000 mA battery. A battery charging circuit is designed to interface a 4.5 x 8.5 inch solar panel with the battery. The second method involve piezoelectric panel to harvest

the vibration on the robot to charge the battery. A panel was designed by another research group at ARRI, consisting of multiple piezoelectric transducers to capture the vibration along the x, y and z directions.

The thesis has contributed in designing a compact, double sided printed circuit board for the second generation ARRI BOT. The design houses interface for crickets, ultrasonic, solar panel, motor, encoders, counters and color sensor with the main Javelin stamp CPU.

The thesis has also contributed in assembling a low cost structural design for the ARRI BOT. The chassis has the provision to house the electronic board, motors, encoders, color sensor, Omni directional wheel, and a solar panel.

The thesis has contributed in testing the performance of localization using active beacons on the second generation ARRI BOT.

Finally, while achieving the above mentioned goals, care is taken to maintain the cost of the low. The current ARRI BOT is approximately \$670, which is comparatively less than other available wheeled robots for research purposes.

## 6.2 Future work

Some of the future work that would expand the horizon of ARRI BOT is mentioned here:

A more extensive analysis on localization needs to be carried out in terms of number of static beacons utilized, positioning of the static beacons to achieve maximum coverage and also testing different formation among the mobile robots to handle the line of sight challenge in the best possible way. One approach would be to have a back to

back facing formation of the robots, such that one set of robots are moving in forward direction while the other set moves in backward direction in order to maintain a line of sight contact with each other at all times.

The next step towards energy harvesting would include exploring different energy harvesting techniques that are suited for indoor application, for instance, a wind mill based energy harvesting to charge the on-board batteries.

In future, effort needs to be taken in achieving de-centralized computing units. The computational capabilities of Javelin stamp and Atmel processor on the cricket could be used to carry out complex calculation and decision making. For instance, the triangulation calculation for position estimation could be taken care by the Javelin stamp or the cricket.

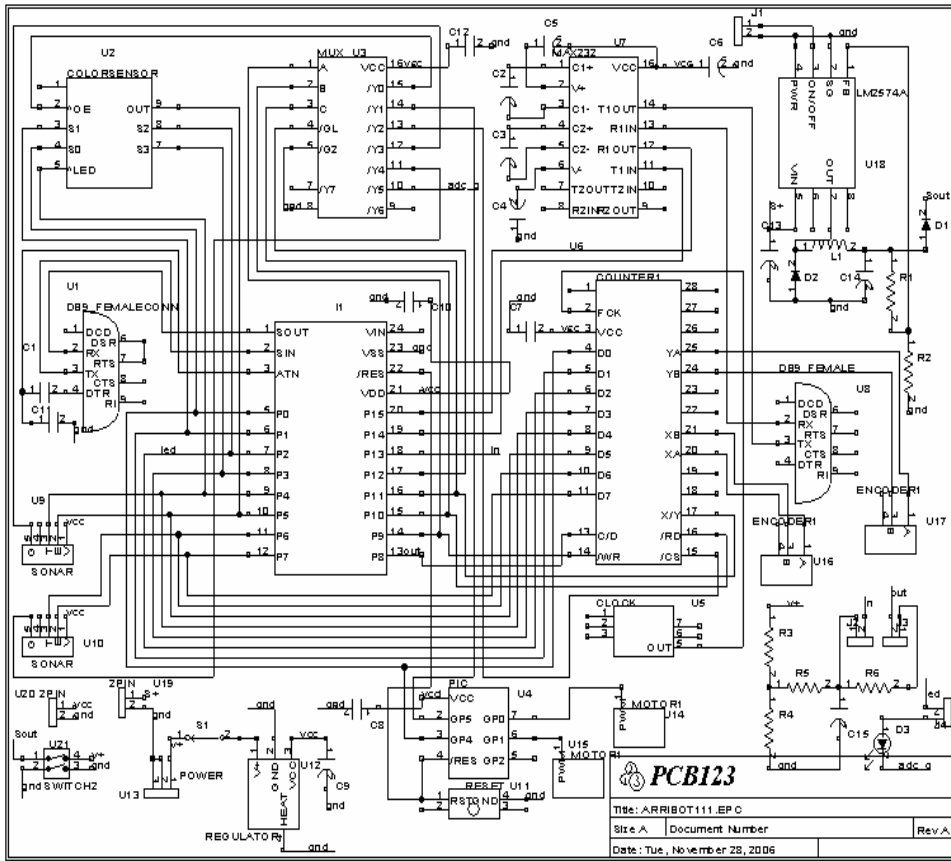
The next step towards the electronic designing would include testing the performance of ARRI BOT using a more powerful CPU. A processor that would provide large memory capacity (over 32KB), additional I/O pins (over 16 pins) and is able to handle interrupts (unlike polling) will aid in handling complex task with minimal programming and easy interfacing with additional sensors, if required.

In future, research can be focused on designing miniaturized, bio-mimetic robots. An autonomous, insect like or flying robots will help in exhibiting robust performance in unstructured environments [67].

## APPENDIX A

### ELECTRICAL DESIGNS OF ARRI BOT

# SCHEMATIC DIAGRAM





The schematic diagram provides a detailed description of the electrical interconnects on the ARRI BOT. The I/O pins on the Javelin Stamp are multiplexed in order to interface multiple peripherals.

<b>Icons</b>	<b>Description</b>
I1	24 pin CPU
U1	DB9 serial connector for CPU
U2	Color sensor TCS230
U3	Multiplexer
U4	PIC
U5	Clock
U6	Counter
U7	MAX 232
U8	DB9 connector for cricket
U9	Ultrasonic range finder
U10	Ultrasonic range finder
U11	Reset push button
U12	5V regulator
U13	Power socket
U14,U15	Motor connection
U16,U17	Optical encoder connector
U18	Switch mode buck regulator
U19	Solar panel socket
U20	Power supply outlet
U21	Switch for battery charging
J1,J2,J3	Jumpers
S1	On/Off switch

## APPENDIX B

### STRUCTURAL DESIGNS OF ARRI BOT

## STRUCTURAL DESIGNS OF ARRI BOT

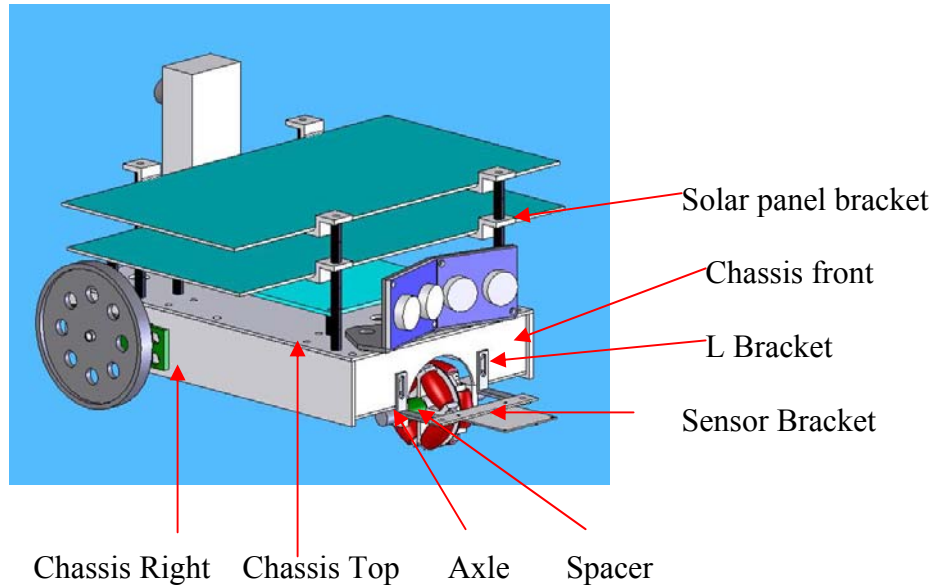


Figure Labeled view of the complete chassis

### *Chassis Top*

The chassis top is approximately 7.5 x 4.7 inches in dimension. Holes are drilled to unify the rest of the chassis parts together and to house the 6 x 3 inches electronic board, ranging modules and brackets for solar panel.

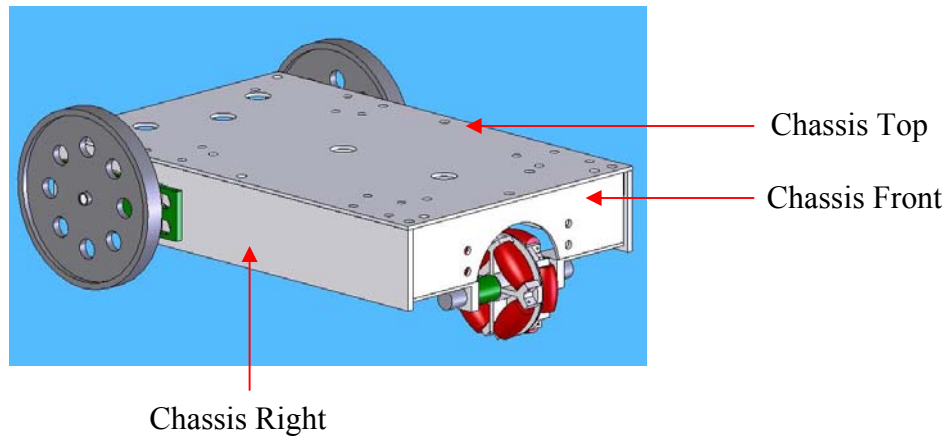


Figure Side view of the chassis

#### *Chassis Front*

Chassis front has a semi-circle cut in the center to house the omni wheel. Holes are drilled to mount the color sensor in a way that will provide 1 inch spacing from the ground.

#### *Chassis Right/Left*

Provision is made to house the servos and the encoder on the right and left side of the chassis

#### *Axle and Spacers*

Axle and spacers are used to fasten the omni wheel with the chassis front firmly.

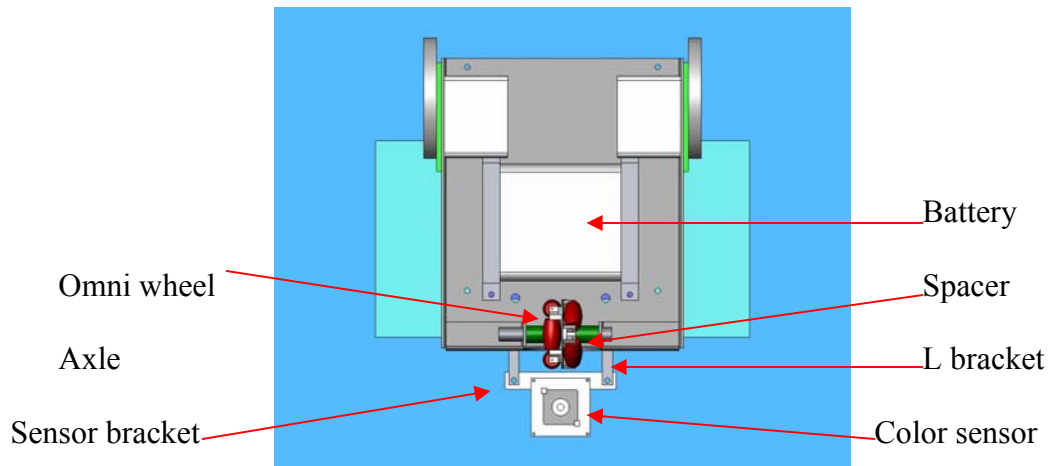


Figure Bottom view of the chassis

#### *L bracket and Sensor bracket*

L bracket and the sensor brackets are used to fasten the color sensor with the chassis front, such that 1 inch spacing is maintained between the sensor and the ground level.

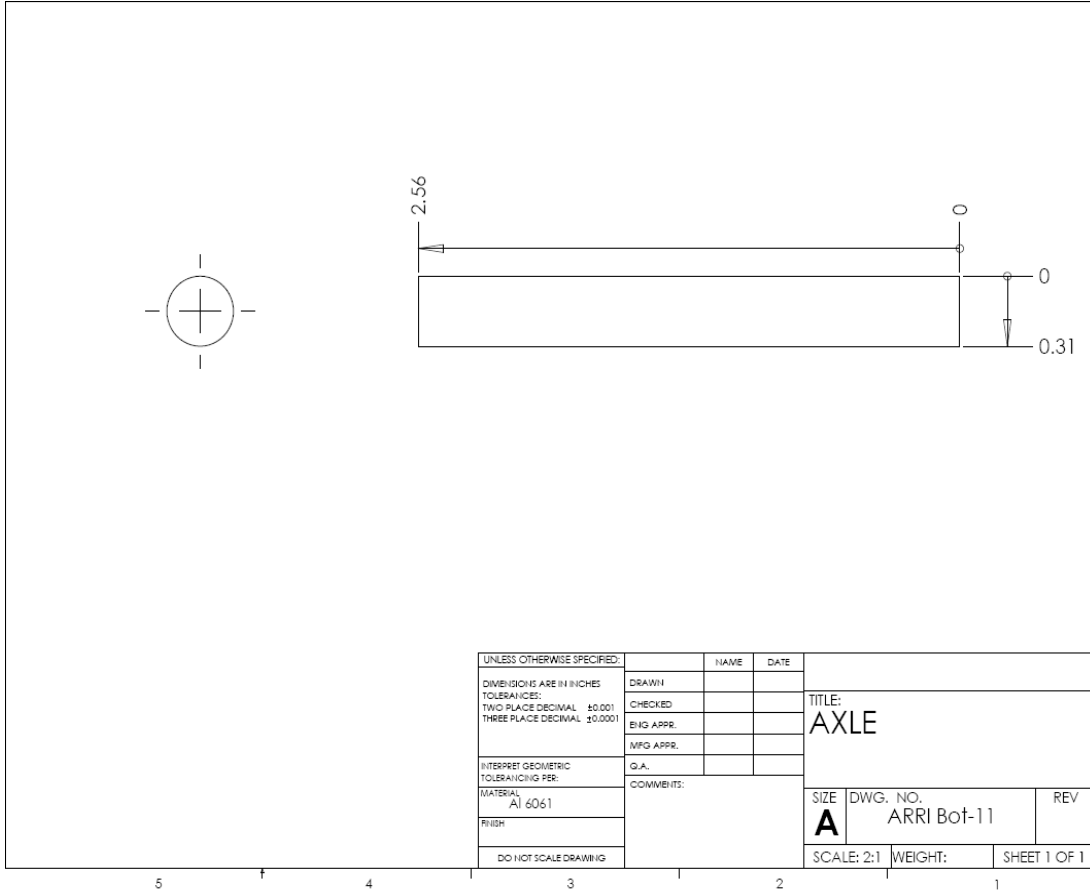
#### *Battery*

Battery is fixed at the bottom of the chassis using removable brackets.

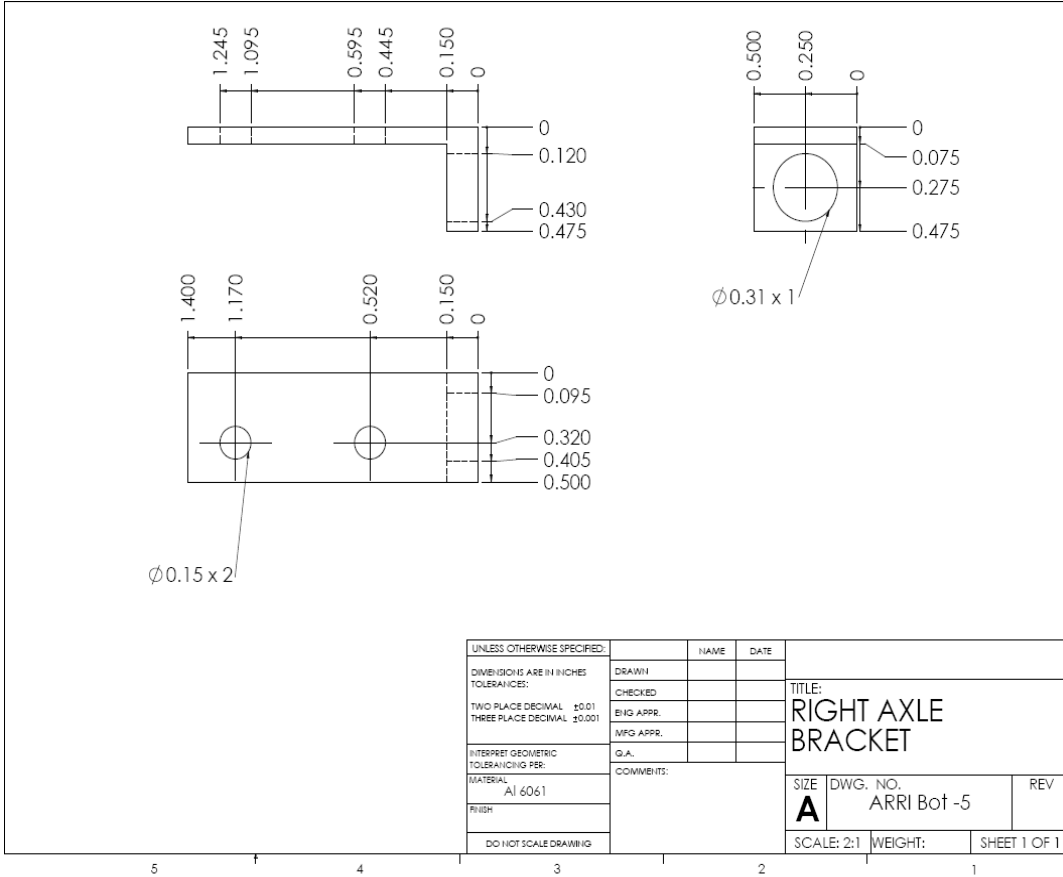
#### *Solar Panel brackets*

Solar panel brackets are designed to house the 4.5 x 8.5 inch solar panel. Standoffs are used to mount the panel as a second level of the chassis. Similar kind of provision is made to add more one more level to the chassis for future developments such as adding a piezoelectric panel for energy harvesting.

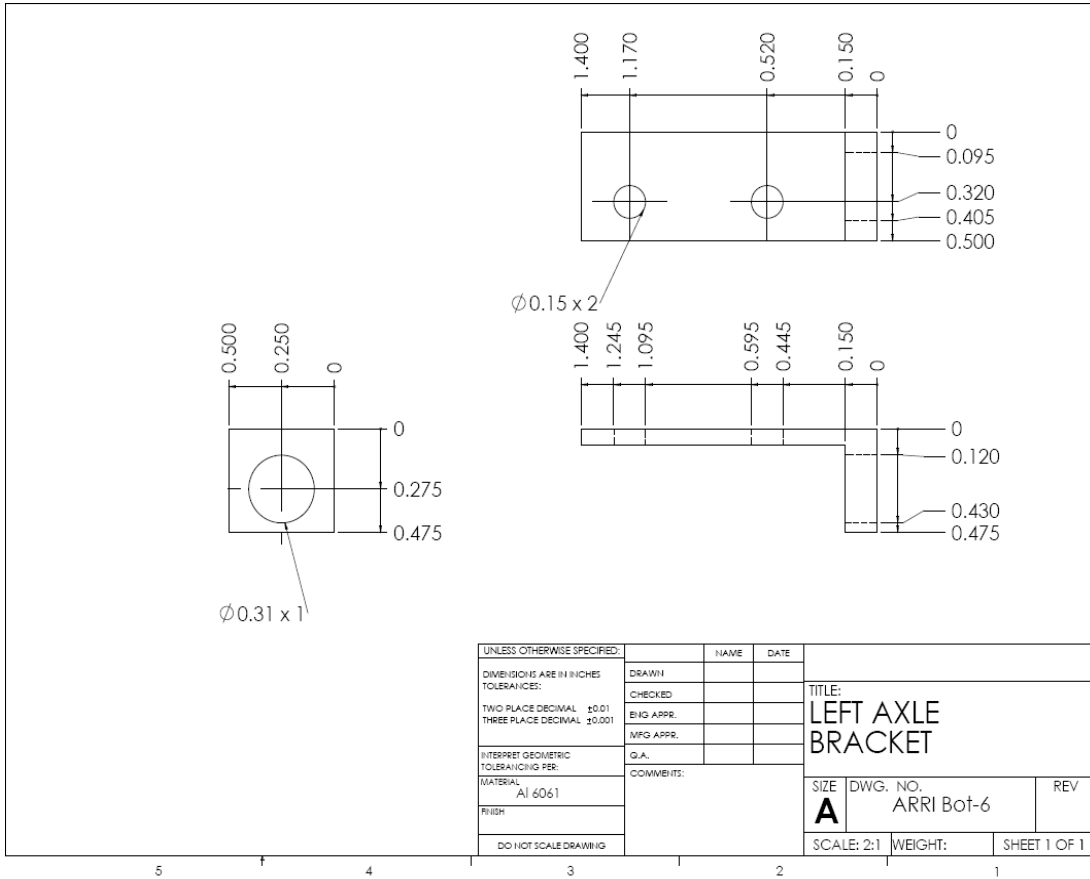
# AXLE



# RIGHT AXLE BRACKET



# LEFT AXLE BRACKET



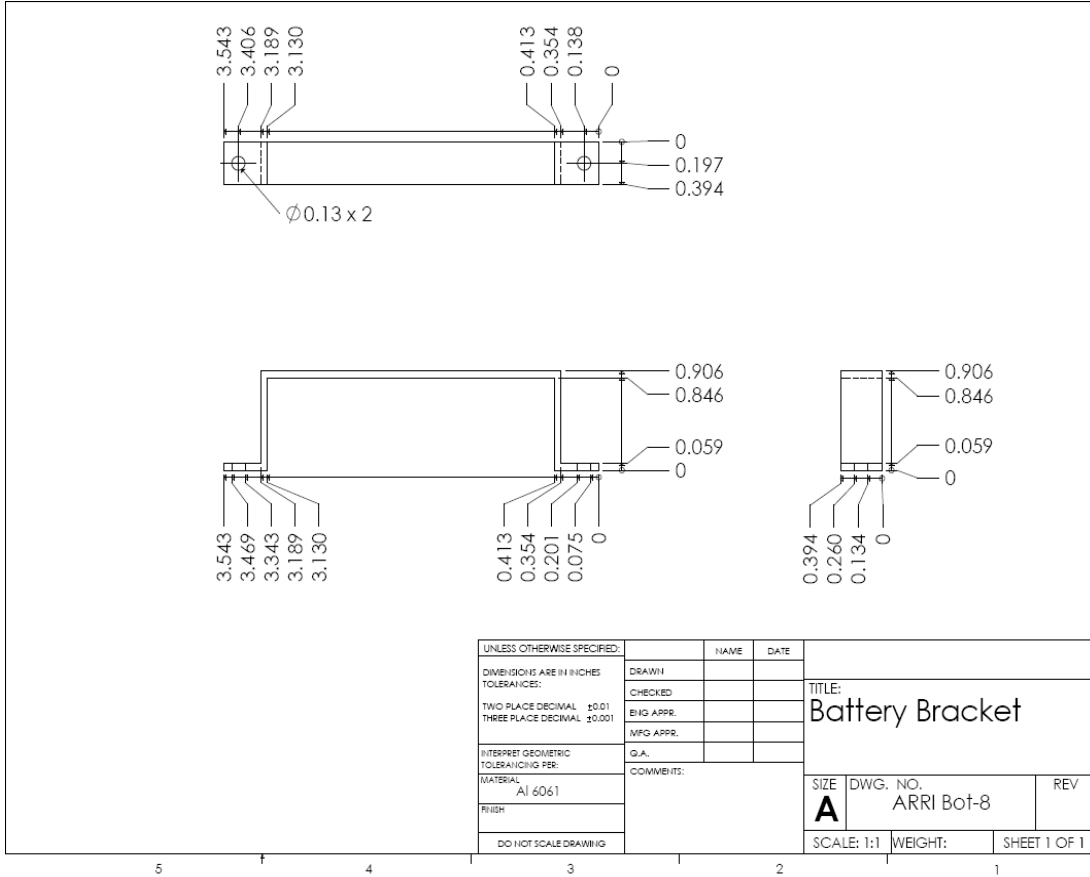
UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES	DRAWN		
TOLERANCES:	CHECKED		
TWO PLACE DECIMAL ±0.01	ENG APPR.		
THREE PLACE DECIMAL ±0.001	MFG APPR.		
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
MATERIAL: Al 6061	COMMENTS:		
FINISH:		SIZE <b>A</b>	DWG. NO. ARRI Bot-6
DO NOT SCALE DRAWING		SCALE: 2:1	WEIGHT: SHEET 1 OF 1

TITLE:  
LEFT AXLE  
BRACKET

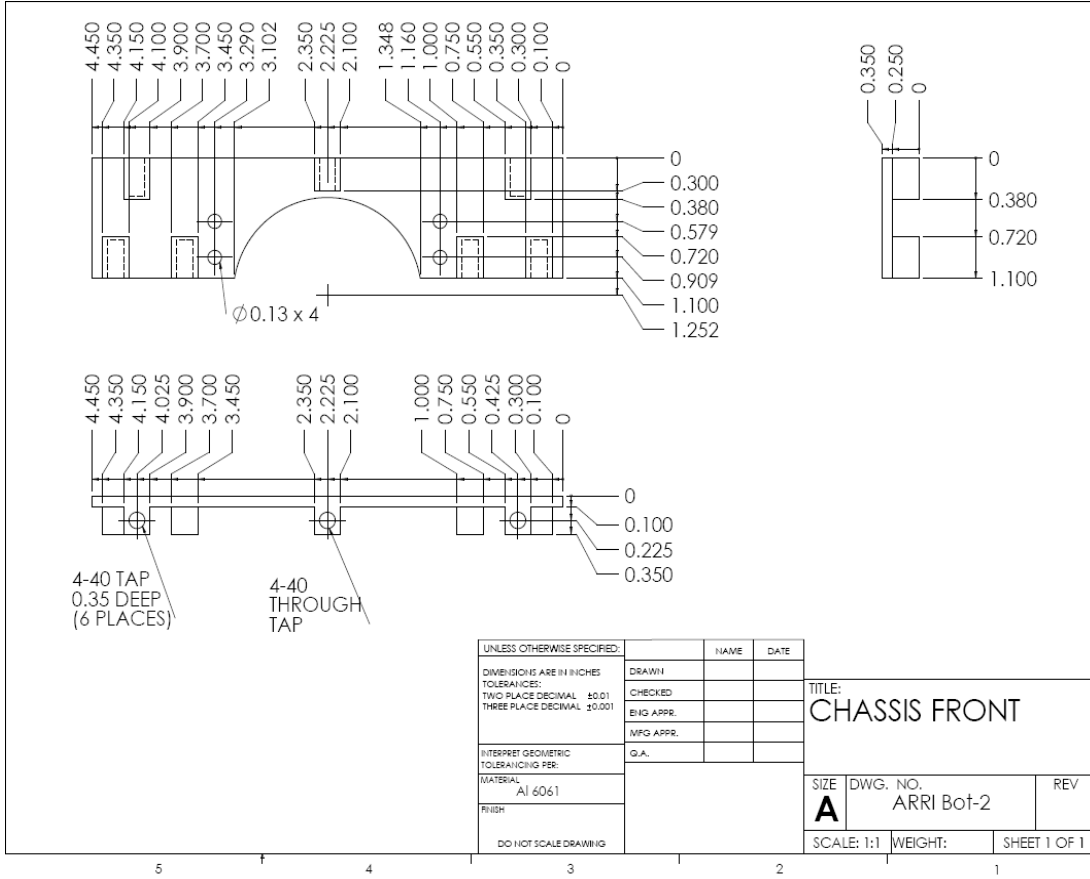
REV



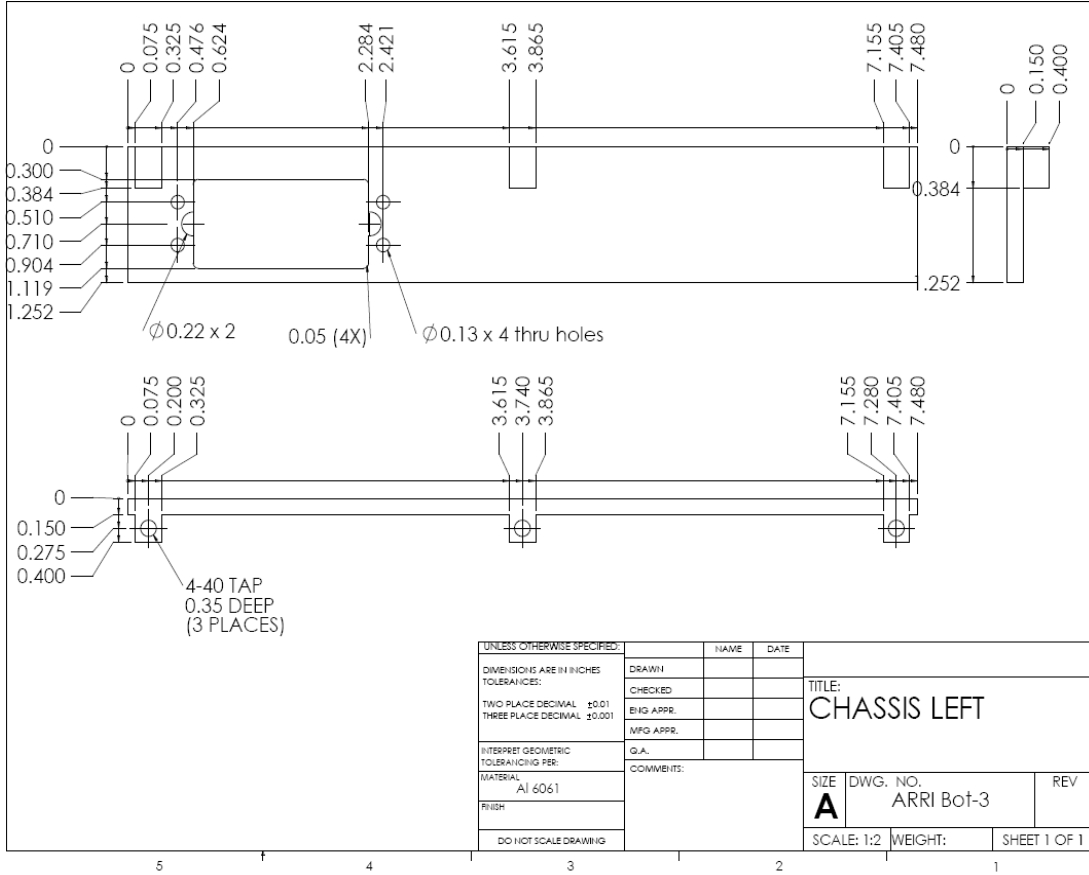
# BATTERY BRACKET



# CHASSIS FRONT



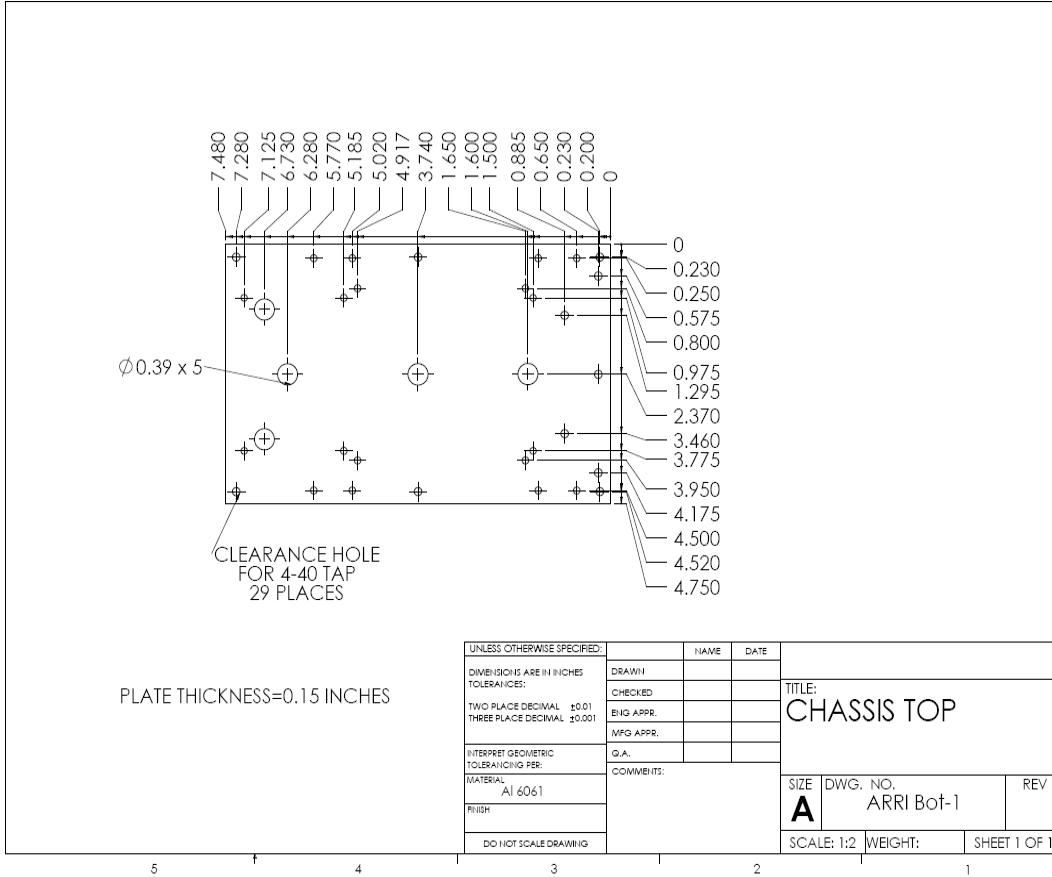
# CHASSIS LEFT



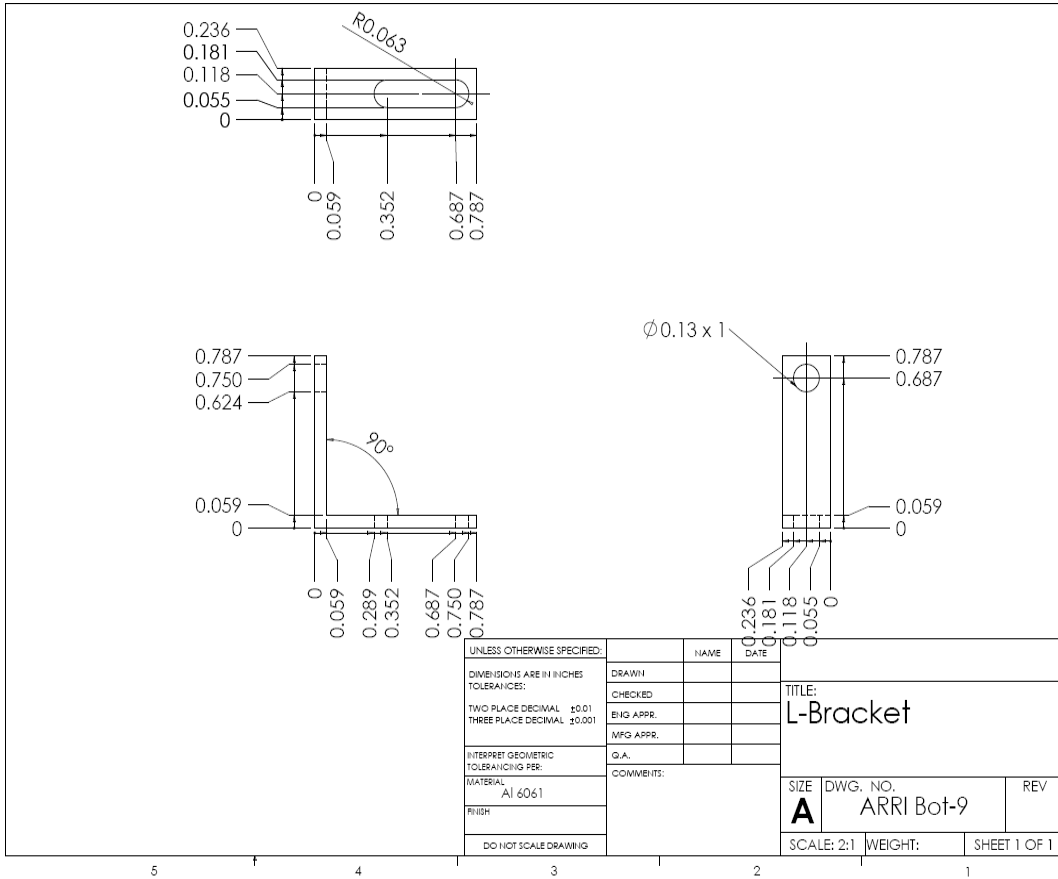
UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES		DRAWN	
TOLERANCES:		CHECKED	
TWO PLACE DECIMAL ±0.01		ENG APPR.	
THREE PLACE DECIMAL ±0.001		MFG APPR.	
INTERPRET GEOMETRIC TOLERANCING PER:		G.A.	
MATERIAL	Al 6061	COMMENTS:	
FINISH		SIZE	DWG. NO.
DO NOT SCALE DRAWING		<b>A</b>	ARRI Bot-3
		SCALE: 1:2	WEIGHT:
			SHEET 1 OF 1



# CHASSIS TOP

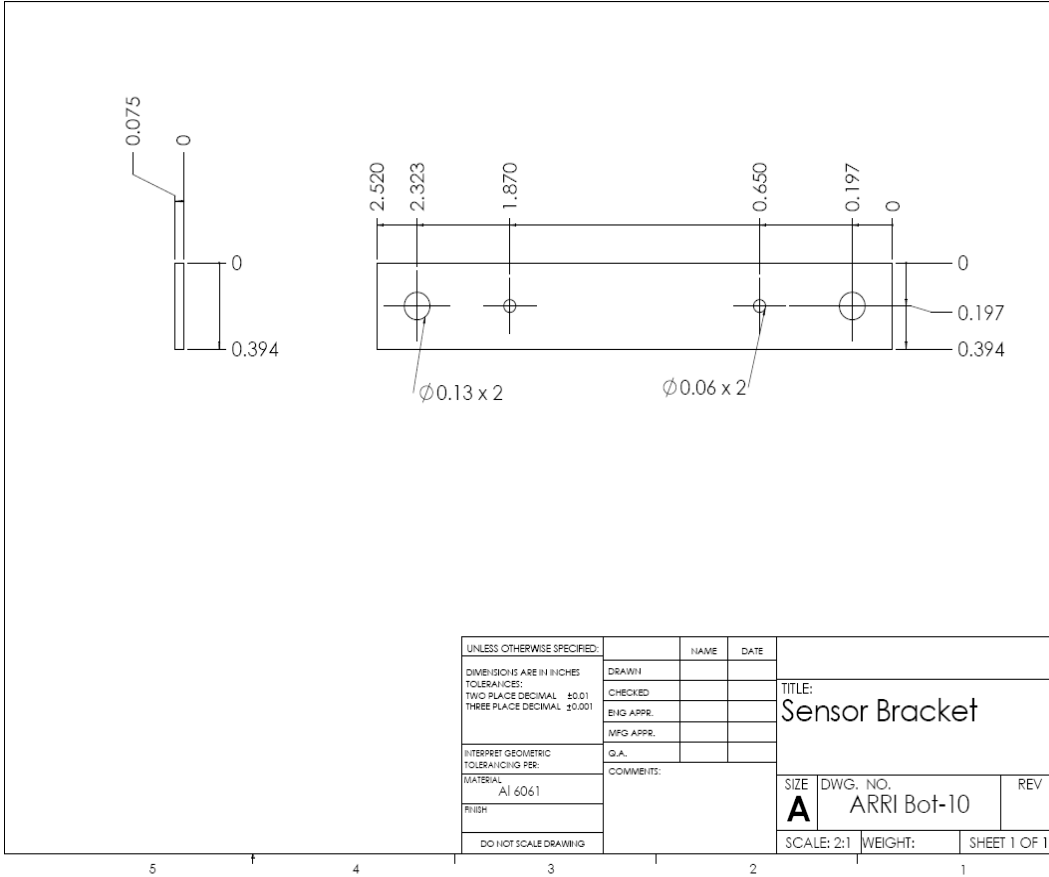


# L-BRACKET

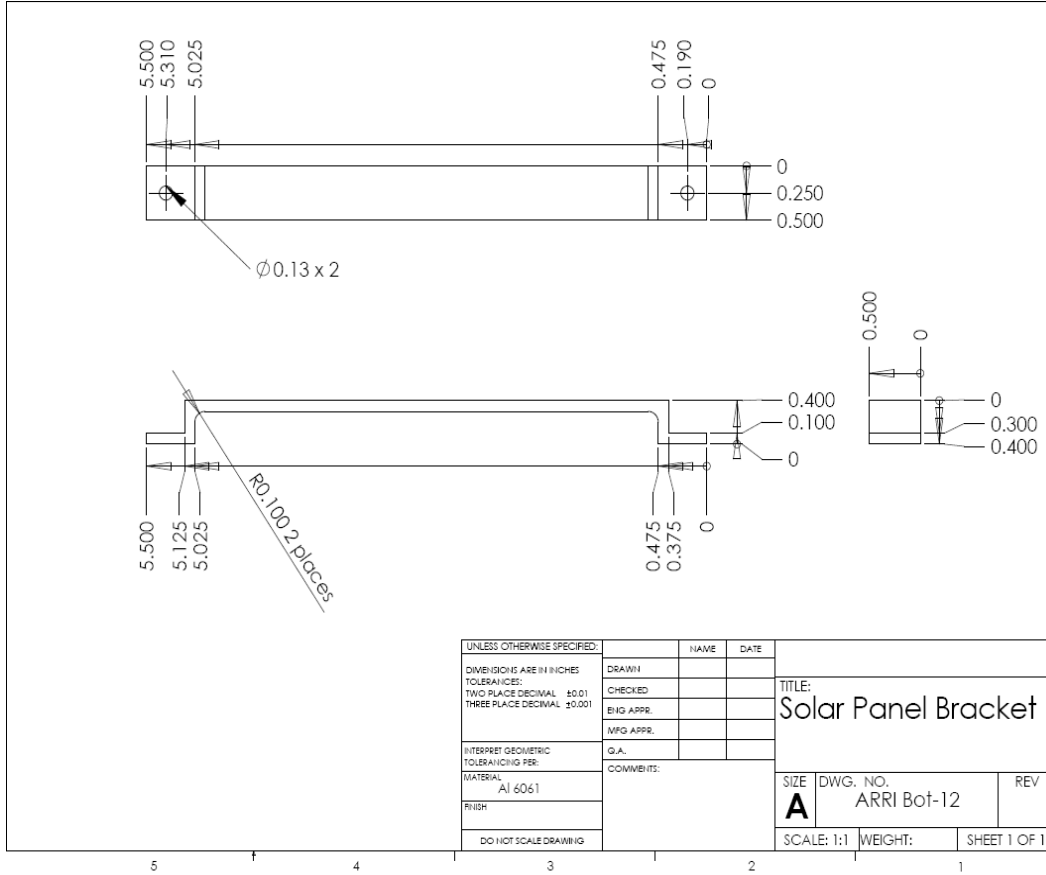


UNLESS OTHERWISE SPECIFIED:		NAME	DATE
DIMENSIONS ARE IN INCHES		DRAWN	
TOLERANCES:		CHECKED	
TWO PLACE DECIMAL ±0.01		ENG APPR.	
THREE PLACE DECIMAL ±0.001		MFG APPR.	
INTERPRET GEOMETRIC TOLERANCING PER:		Q.A.	
MATERIAL: Al 6061		COMMENTS:	
FINISH:		SIZE DWG. NO. <b>A</b> ARRI Bot-9	
DO NOT SCALE DRAWING		SCALE: 2:1	WEIGHT: SHEET 1 OF 1
		REV	

# SENSOR BRACKET

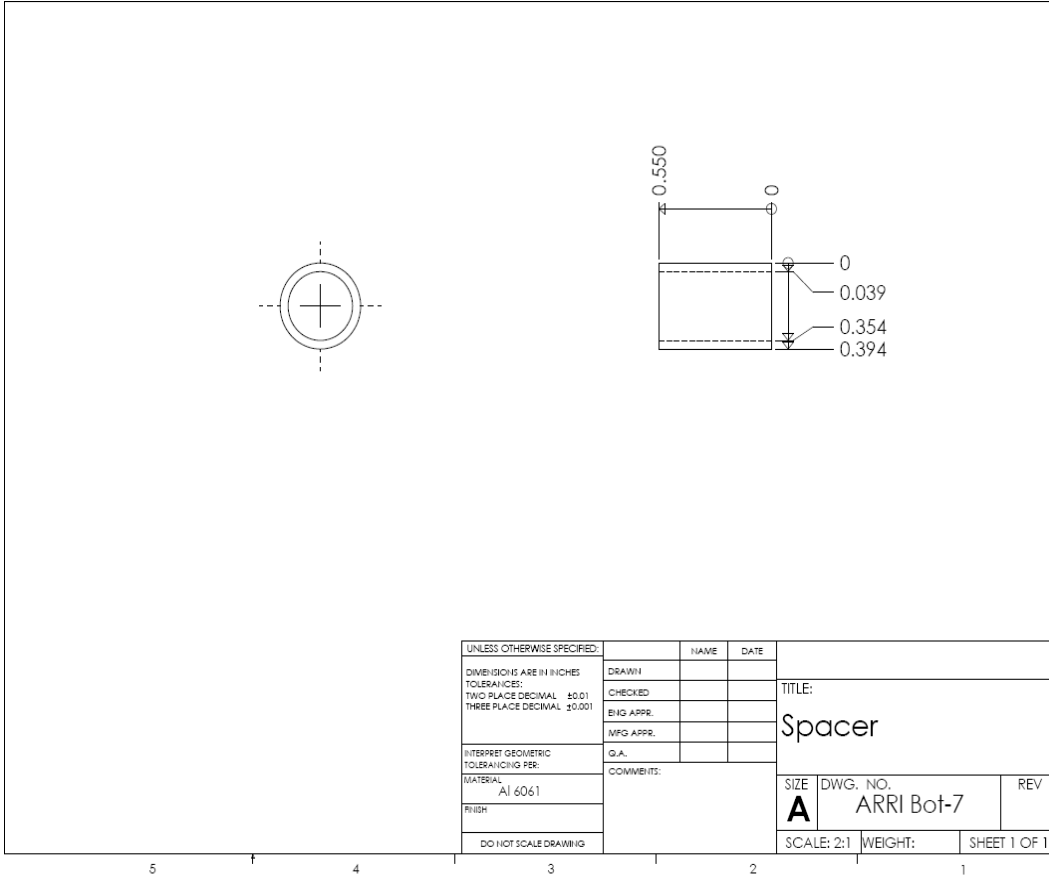


# SOLAR PANEL BRACKET





# SPACER



UNLESS OTHERWISE SPECIFIED:		NAME	DATE		
DIMENSIONS ARE IN INCHES	DRAWN			TITLE:	
TOLERANCES:	CHECKED			Spacer	
TWO PLACE DECIMAL ±0.01	ENG APPR.			SIZE DWG. NO.	
THREE PLACE DECIMAL ±0.001	MFG APPR.			A ARRI Bot-7	
INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.			REV	
MATERIAL: Al 6061	COMMENTS:			SCALE: 2:1	
FINISH:				WEIGHT:	
DO NOT SCALE DRAWING				SHEET 1 OF 1	

## REFERENCES

- [1] Deepak Ganesan, Sylvia Ratnasamy, Hanbiao Wang, Deborah Estrin, “Coping with irregular spatio-temporal sampling in sensor networks”, in *Second Workshop on Hot Topics in Networks (HotNets-II) 2003*.
- [2] Andrew Howard, Maja J Matarić and Gaurav S Sukhatme, “An Incremental Self-Deployment Algorithm for Mobile Sensor Networks”, in *Autonomous Robots, Special Issue on Intelligent Embedded Systems*.
- [3] Wikipedia, [http://en.wikipedia.org/wiki/Wireless\\_sensor\\_network..](http://en.wikipedia.org/wiki/Wireless_sensor_network..)
- [4] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”, in *Proceedings of the 33rd Hawaii International Conference on System Sciences 2000*.
- [5] Frank Dellaert, Dieter Fox, Wolfram Burgard, Sebastian Thrun, “Monte Carlo Localization for Mobile Robots”, in *IEEE International Conference on Robotics and Automation (ICRA99), May, 1999*.
- [6] Wikipedia “Robots”, <http://en.wikipedia.org/wiki/Robots><http://>
- [7] G. T. Sibley, M. H. Rahimi, G. S. Sukhatme, "Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks", in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2002)*.
- [8] MOBILEROBOTS, <http://www.activrobots.com/ROBOTS/amigobot.html>.
- [9] MIT's Yuppy, <http://www.ai.mit.edu/projects/mobile-robots/robots.html#yuppy>.

- [10] Acroname Inc, <http://www.acroname.com/garcia/garcia.html>.
- [11] TagBot Demo Guide (PDF), <http://sparkyb.net/temp/staytondoc/tagproject>.
- [12] T. A. Dahlberg, A. Nasipuri, C. Taylor, “Explorebots: A Mobile Network Experimentation Testbed”, in *SIGCOMM’05 Workshops, August 22–26, 2005*.
- [13] M. Michałek “Tracking controller for nonholonomic steering system with velocity constraints”, in *The 7th IFAC Symposium on Robot Control (SYROCO’03), 1-3 September 2003, Wrocław, Poland, pp. 587-592*.
- [14] W. Kowalczyk, K. Kozłowski “Coordinate strategy for formation of mobile robots”, in *The 9th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR’03), pp. 789-796*.
- [15] Janne Haverinen, Mikko Parpala and Juha Rönning, “A Miniature Mobile Robot With a Color Stereo Camera System for Swarm Robotics Research”, in *International Conference on Robotics and Automation Barcelona, Spain, April 2005*.
- [16] Wikipedia, [http://en.wikipedia.org/wiki/Sensor\\_network](http://en.wikipedia.org/wiki/Sensor_network).
- [17] Jeff Thorn, “Deciphering TinyOS Serial Packets”, Octave Tech Brief #5-01.
- [18] Arnab Chakrabarti, Ashutosh Sabharwal, Behnaam Aazhang, “MultiHop Communication is Order Optimal for Homogeneous Sensor Networks” in *IPSN’04, April 26–27, 2004, Berkeley, California, USA*.
- [19] Thiemo Voigt, Adam Dunkels, Torsten Braun, “On-demand Construction of Non-interfering Multiple Paths in Wireless Sensor Networks”, in *2nd Workshop on Sensor Networks at Informatik 2005*.

- [20] Bhaskar Krishnamachari, Deborah Estrin, Stephen Wicker, “Modelling Data-Centric Routing in Wireless Sensor Networks”, in *IEEE INFOCOM 2002*.
- [21] P. Berthou, T. Gayraud, O. Alphand, C. Prudhommeaux, M. Diaz  
“A Multimedia Architecture for 802.11b networks”, in *Wireless Communications and Networking, 2003(WCNC)*, on page(s): 1742- 1747 vol.3.
- [22] How stuff works, <http://www.nasm.si.edu/gps/work.html>.
- [23] Wikipedia , <http://en.wikipedia.org/wiki/GPS>.
- [24] Rudy Negenborn, Thesis, “Robot Localization and Kalman Filters”.
- [25] Antonio Sgorbissa, Maurizio Piaggio, Renato Zaccaria, “Autonomous Navigation and Localization in Service Mobile Robotics”, in *Intelligent Robots and Systems, 2001*, on page(s): 2024-2029 vol.4.
- [26] Chris Savarese, Jan M. Rabaey, Jan Beutel, “LOCATIONING IN DISTRIBUTED AD-HOC WIRELESS SENSOR NETWORKS”, in *ICASSP 2001*.
- [27] Toshihiro Aono, Kenjiro Fujii, Shintaro Hatsumoto, Takayuki Kamiya,  
“Positioning of vehicle on undulating ground using GPS and dead reckoning”, in *Proceedings of the 1998 IEEE International Conference on Robotics & Automation Leuven, Belgium May 1998*.
- [28] T. Schönberg, M. Ojala, J. Suomela, A. Torpo, A. Halme, “Positioning an Autonomous off-road vehicle by using fused DGPS AND Inertial navigation”, Helsinki University of Technology, Automation Technology Laboratory.
- [29] Andreas Birk and Cosmin Condea, “Mobile Robot Communication without the Drawbacks of Wireless Networking”, in *RoboCup 2005*, pp. 585 592.

- [30] Andrew M. Ladd, Kostas E. Bekris, Algis Rudys, “Robotics-Based Location Sensing using Wireless Ethernet”, in *MOBICOM’02, September 23–26, 2002, Atlanta, Georgia, USA*.
- [31] Chee-Yee Chong, Srikanta P. Kumar, “Sensor networks: Evolution, opportunities, and challenges”, in *The proceedings of IEEE, vol. 91, no. 8, August 2003*.
- [32] Zhigang Wang, MengChu Zhou and Ninvan Ansari, “Ad-hoc Robot Wireless Communication”, in *Systems, Man and Cybernetics, 2003, on page(s): 4045- 4050, vol4*.
- [33] G M Lundy and A J Bekas, “WIRELESS COMMUNICATIONS FOR A MULTIPLE ROBOT SYSTEM”, in *MILCOM 97, on page(s): 1293-1297 vol.3*.
- [34] Maurizio Piaggio, Antonio Sgorbissa, Renato Zaccaria, “Autonomous Navigation and Localization in Service Mobile Robotics” in *Intelligent Robots and Systems, 2001, on page(s): 2024-2029 vol.4*.
- [35] Dan O. Popa, Arthur C. Sanderson, Rick Komerska, Sai Mupparapu, Richard Blidberg and Steven Chappel, “Adaptive Sampling Algorithms for Multiple Autonomous Underwater Vehicles”, in *Proc. of 2004 Workshop on Underwater Vehicles, Sebasco Estates, ME, June 2004*.
- [36] Mohammad Rahimi, Richard Pon, William J. Kaiser, Gaurav S. Sukhatme, Deborah Estrin. and Mani Srivastava, “Adaptive Sampling for Environmental Robotics”, in *The 2004 IEEE International Conference on Robotics and Automation (ICRA)*.
- [37] Toshihiro Aono, Kenjiro Fujii, Shintaro Hatsumoto, Takayuki Kamiya,

“Positioning of vehicle on undulating ground using GPS and dead reckoning”, in *Proceedings of the 1998 IEEE International Conference on Robotics & Automation Leuven, Belgium May 1998*.

[38] SeattleRobot Society, <http://www.seattlerobotics.org/encoder/200009/S3003C.html>

[39] Hyper Physics, <http://hyperphysics.phy-astr.gsu.edu/hbase/hframe.html>.

[40] MASSA, <http://www.massa.com/fundamentals.htm>.

[41] Parallax, <http://www.acroname.com/robotics/parts/R93-SRF04p.pdf>.

[42] Xiaoli Mu, “Exploiting Cooperative Diversity for Wireless Sensor Networks Using Medium Access Control”, in *MILCOM 2004 - 2004 IEEE Military Communications Conference*.

[43] Huffman code, “<http://michael.dipperstein.com/huffman/index.html>”.

[44] “<http://www.k-team.com/kteam/index.php?site=1&rub=3&upPage=200&page=197&version=EN>”.

[45] G. Caprari, K. O. Arras, R. Siegwart, “The Autonomous Miniature Robot Alice: from Prototypes to Applications”, in *Intelligent Robots and Systems, 2000*, on page(s): 793-798 vol.1.

[46] Polybot, <http://www2.parc.com/spl/projects/modrobots/polybot/polybot.html>.

[47] Koushil Sreenath, “Adaptive sampling with mobile wireless sensor network”, a Master’s thesis at University of Texas at Arlington, 2005.

[48] Chad F. Helm, “Robotic sensor deployment using potential fields”, a Master’s thesis at Rensselaer Polytechnic Institute, 2004.

[49] Parallax, TCS230 <http://www.parallax.com/dl/docs/prod/datast/TCS230.pdf>.

- [50] Acroname, <http://www.acroname.com/robotics/parts/R93-SRF04.html>.
- [51] National semiconductor, “LM2574/LM2574HV Series SIMPLE SWITCHER™ 0.5A Step-Down Voltage Regulator”.
- [52] Crossbow, <http://cricket.csail.mit.edu/>.
- [53] Crossbow, “[http://www.xbow.com/Support/Support\\_pdf\\_files/MPR-MIB\\_Series\\_Users\\_Manual.pdf](http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf)”.
- [54] SCARA, [http://en.wikipedia.org/wiki/SCARA\\_robot](http://en.wikipedia.org/wiki/SCARA_robot).
- [55] PUMA 500, [http://www-de.ksc.nasa.gov/dedev/admin/org/mm-g3/MAIN\\_PAGE/AUTOMATION\\_MAIN\\_PAGE/PROJECTS/PUMA/PUMA.HTML](http://www-de.ksc.nasa.gov/dedev/admin/org/mm-g3/MAIN_PAGE/AUTOMATION_MAIN_PAGE/PROJECTS/PUMA/PUMA.HTML).
- [56] Spectrum of light, <http://www.uvguide.co.uk/>.
- [57] Solar panel, <http://www.plastecs.com/>.
- [58] PCB123, <http://www.pcb123.com/>
- [59] Mohammad Rahimi, Richard Pon, William J. Kaiser, Gaurav S. Sukhatme, Deborah Estrin and Mani Srivastava, “Adaptive Sampling for Environmental Robotics”, in *The IEEE International Conference on Robotics and Automation (ICRA) 2004*.
- [60] D. Popa, et.al.,”Optimal Sampling using Singular Value Decomposition of the Parameter Variance Space”, in *The proceedings of IROS '05, Edmonton, Canada, August 2005*.
- [61] A. C. Sanderson, “Multirobot navigation using cooperative teams”, in *Distributed Autonomous Robotic Systems 2*, pp. 389-400, 1998.

- [62] D. Popa, C. Helm, H. E. Stephanou, A. Sanderson, “Robotic Deployment of Sensor Networks using Potential Fields”, in *Proc. Of International Robotics and Automation Conference, April-May 2004*.
- [63] F. L. LEWIS “Wireless Sensor Networks”, in *Smart Environments: Technologies, Protocols and Applications, New York, 2004*.
- [64] Parallax, [http://www.robotstore.com/download/transceiver\\_Manual\\_433MHz.pdf](http://www.robotstore.com/download/transceiver_Manual_433MHz.pdf)
- [65] Wikipedia, [http://en.wikipedia.org/wiki/Humanoid\\_robot](http://en.wikipedia.org/wiki/Humanoid_robot).
- [66] Robovie-M, <http://www.irc.atr.jp/productRobovie/robovie-m-e.html>.
- [67] Bio-mimetic robotics, <http://www-cdr.stanford.edu/biomimetics/>.



## BIOGRAPHICAL INFORMATION

Jaymala S. Ghadigaonkar received her Bachelor of Engineering degree in Electronics from Pune University, India in 2003. She then joined in Masters Program in Electrical Engineering at University of Texas at Arlington in fall 2004. She was a graduate teacher's assistant for the Semiconductor Electronics Design course in spring 2006. Motivated by the interest in the field of system controls and robotics, she joined the Automation & Robotics Research Institute (ARRI) and was a graduate research assistant in fall 2006, under the able supervision of Dr. Dan O. Popa. During her tenure she worked on the designing a low cost mobile wireless sensor node. This thesis is reflects her work on research study and implementation done on this wireless sensor node.