Winter 12-18-2017

ROS-Enabled Framework for a Miniature Hexapod as a Mobile Robot Research Platform

Joseph Michael Kloeppel
University of New Mexico

Follow this and additional works at: https://digitalrepository.unm.edu/ece_etds

Part of the Controls and Control Theory Commons

Recommended Citation

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact disc@unm.edu.
Joseph Kloeppel  
Candidate  

Electrical & Computer Engineering  
Department  

This thesis is approved, and it is acceptable in quality and form for publication:  

Approved by the Thesis Committee:  

Dr. Rafael Fierro, Chairperson  

Dr. Melanie Moses  

Dr. Francesco Sorrentino
ROS-Enabled Framework for a Miniature Hexapod as a Mobile Robot Research Platform

by

Joseph Kloeppel

B.S., New Mexico Institute of Mining and Technology, 2015

THESIS

Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico
Albuquerque, New Mexico

May, 2018
Dedication

To my parents, Michael and Camilla, and my dearest Jasmin, for always providing me with love, support, and encouragement.

"It’s a matter of faith." – Vladimir
Acknowledgments

I would like to thank my academic advisor and mentor, Dr. Rafael Fierro, for giving me the opportunity to work with him in the MARHES laboratory and for providing insight, direction, and support. Additionally, I would like to thank Dr. Melanie Moses and Dr. Francesco Sorrentino for being a part of my thesis committee and for being memorable and invaluable teachers throughout my graduate studies.

I would like to thank my brother, Jacob, and my sister, Alexandria, for their support and encouragement.

Lastly, I would like to thank all of my friends from the MARHES laboratory that were instrumental to the success of my graduate studies and research. I would like to say thank you to Dr. Jonathan West, Shakeeb Ahmad, Steve Maurice, Christoph Hintz, Gregory Brunson, Carolina Gomez, Rebecca Kreitinger, and all of the other students which were a part of the MARHES team.

This work was supported in part by the Army Research Lab Micro Autonomous Systems and Technology Collaborative Alliance (ARL MAST-CTA #W911NF-08-2-0004)
ROS-Enabled Framework for a Miniature Hexapod as a Mobile Robot Research Platform

by

Joseph Kloeppel

B.S., New Mexico Institute of Mining and Technology, 2015
M.S., Electrical Engineering, University of New Mexico, 2018

Abstract

The purpose of this thesis is to investigate and develop a framework for connected six-legged robots which can be used as a Robotic Operating System (ROS) based research platform. The research presented aims to purvey the necessary engineering and scientific steps needed to evolve a consumer-grade connected toy into a fully functioning and highly capable robotic system. Such a platform can be used to simulate and implement novel biologically inspired swarm research. Crawling robots have the advantage of being able to scale terrains which wheeled mobile robots may not and possess many interesting characteristics.

The miniROaCH, miniature ROS-enabled and Crawling Hexapod, is a lightweight and small-scale robot which is able to navigate and traverse unknown and uneven terrain. The miniROaCH robot was designed to utilize both commercially accessible components and 3D printable parts. With a 1GHz single-core on-board processor and an 8MP high-definition camera, the platform allows for situational awareness, object avoidance and mapping capabilities. Furthermore, each miniROaCH is capable of

iv
running either centralized or distributed control algorithms, making the platform an excellent choice for cost-effective swarm robotic research.

This thesis presents a system’s architecture on both a hardware and software level. A low-level velocity controller is implemented and the results are shown. Additionally, a high-level controller has been used to provide position control for the system. Using internal feedback from on-board sensors and visual landmarks within the environment, the miniROaCH robot is able to estimate its position and orientation within the world frame of reference. This allows the miniROaCH to navigate environments without the need for external sensors, providing the robot with a high level of autonomy. The development of the miniROaCH is documented and applications are explored.
Contents

List of Figures x

List of Tables xiv

Glossary xv

1 Introduction 1

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Related Work 3

1.3.1 Crawling Hexapods 3

1.3.2 Related Testbeds 6

1.3.3 Robotic Exploration 8

1.4 Contributions 8

1.5 Organization of Thesis 9
2 System Overview

2.1 Hardware .................................................. 10
  2.1.1 miniROaCH ............................................. 10
  2.1.2 Physical Build ....................................... 11
  2.1.3 Electronics ........................................... 13
  2.1.4 Communication Network .............................. 17
  2.1.5 Vicon System .......................................... 18

2.2 Software ................................................ 19
  2.2.1 ROS ..................................................... 19

3 Modeling and System Architecture 25

3.1 Kinematics Model ........................................ 25

3.2 Dynamic Model .......................................... 28
  3.2.1 Actuator Dynamics ................................... 32

3.3 System Architecture ..................................... 35

4 Control Design, Localization, and Implementation 37

4.1 Controller Design ........................................ 37

4.2 State Estimator ........................................... 40

4.3 Localization ............................................... 42

4.4 Hardware Implementation ............................... 44
Contents

D CAD Model 82

E Bill of Materials (BOM) 83

References 84
List of Figures

2.1 miniROaCH robot. .............................................. 11
2.2 miniROaCH side view. ......................................... 12
2.3 Kamigami toy (top view). ................................. 13
2.4 Kamigami toy (side view). ................................. 13
2.5 Raspberry Pi Zero W [1]. ................................. 14
2.6 INA219 current sensor. ...................................... 14
2.7 BNO055 9-DOF IMU. ........................................... 15
2.8 Serial Controlled Motor Driver. ......................... 15
2.9 Pimoroni LiPo SHIM available from [2]. ............. 16
2.10 miniROaCH top view with all system components attached. .............................. 17
2.11 Kamigami robot with retro-reflective markers within the Vicon motion capture area in the MARHES Laboratory. ............................ 19
2.12 ROS Node/Topic structure [3]. ......................... 21
2.13 Example catkin workspace. ............................. 23
List of Figures

3.1 Differential drive mobile robot. ........................................... 27
3.2 miniROaCH dynamic model diagram. .................................... 28
3.3 DC motor diagram with current sense resistor. ....................... 32
3.4 Two 6 mm coreless brushed DC motors with a 16:1 compound gear reduction. .......................................................... 34
3.5 Overview of the inner and outer closed loop system architecture implemented on the miniROaCH. ................................. 35
4.1 Low-level controller block diagram. ..................................... 38
4.2 High-level controller block diagram. .................................... 39
4.3 $\psi_{ref} = \text{heading}_\text{func}(x_{ref}, y_{ref}, \hat{x}_{ekf}, \hat{y}_{ekf})$. .......................................................... 40
4.4 Tag36h11 style AprilTag. .................................................. 42
4.5 TF (Transform) tree structure. .......................................... 43
4.6 Velocity Control Plot ...................................................... 46
4.7 Position plots which shows the path a miniRoaCH robot follows. The red circles represent a target region of success for the robot. .... 47
4.8 miniROaCH within the testbed environment. AprilTag are positioned within the environment as known landmarks to provide visual odometry. ......................................................... 48
4.9 TF connections which link the miniROaCH to a tag within the miniROaCH’s camera view and then to the map frame. The robot can be seen within the physical environment in Figure 4.8. ...... 48
## List of Figures

4.10 Heading plot comparing ground truth measurements (Vicon), IMU data (Heading), and the output of the extended kalman filter (EKF). 49

4.11 Another heading plot comparing ground truth measurements (Vicon), IMU data (Heading), and the output of the extended kalman filter (EKF). 49

5.1 This figure demonstrates an example scenario in which a MAST system can be utilized. 52

5.2 This figure demonstrates an example MAST scenario shown in the Vicon area within the MARHES Laboratory. Two quadrotors fly above a swarm of miniROaCHs and a Turtlebot. Four quadrotors can be seen on the floor on the right. 53

5.3 This figure demonstrates an example scenario in which a MAST system can be utilized. 54

5.4 Generated 3D map overlay on an image of the actual environment. 56

5.5 Generated 3D map. 56

5.6 An example scenario with an overhead view of the tabletop testbed with 3 miniROaCH robots avoiding obstacles. 58

5.7 Snapshots of a quadrotor’s camera view while 3 miniROaCHs execute a follow-the-leader controller in the miniROaCH Virtual Testbed. The robots are crawling in a clockwise rotation. 59

5.8 miniROaCH Virtual Testbed simulation in Gazebo (3 miniROaCH ground agents and one hovering Hummingbird aerial vehicle). 62
List of Figures

5.9  Top Plot: x and y position data w.r.t. the world reference frame of the Leader miniROaCH. Middle Plot: x and y position data of the first Follower miniROaCH w.r.t. the Leader miniROaCH’s reference frame. Bottom Plot: x and y position data w.r.t. the first Follower miniROaCH. .................................................. 63

B.1  mRB: miniROaCH Board (PCB). ......................... 75
B.2  mRB: miniROaCH Board (PCB) 3D view. ............... 76

C.1  Original miniROaCH. ........................................... 77
C.2  miniROaCH: REV A. ............................................. 78
C.3  miniROaCH: REV B. ............................................. 79
C.4  miniROaCH: REV C. ............................................. 79
C.5  miniROaCH: REV D. ............................................. 80
C.6  miniROaCH: REV E, current revision. .................... 80

D.1  3D CAD model rendering of the miniROaCH robot with the PCB, Pi Zero, and camera. ......................... 82
List of Tables

E.1 miniROaCH Bill of Materials. ............................. 83
Glossary

\{W\}  World frame

\{B\}  Body frame

\(x\), \(y\), and \(z\)  Position of the miniROaCH in world frame

\(\theta\), \(\phi\), and \(\psi\)  miniROaCH Euler angles in world frame

\(\psi\)  miniROaCH heading in world frame

\(\hat{x}\), \(\hat{y}\), and \(\hat{z}\)  Position estimates of the miniROaCH in world frame

\(\hat{\theta}\), \(\hat{\phi}\), and \(\hat{\psi}\)  miniROaCH Euler angle estimates in world frame

\(\hat{x}_{ekf}\) and \(\hat{y}_{ekf}\)  Extended Kalman Filter estimates of the miniROaCH position in world frame

\(\hat{\psi}_{ekf}\)  Extended Kalman Filter estimate of the miniROaCH heading in world frame

\(x_{ref}\) and \(y_{ref}\)  Input reference position

\(\psi_{ref}\)  Calculated reference heading

\(v\)  Linear velocity

\(w\)  Angular velocity
Glossary

\( w_l \)  
Left legs angular velocity

\( w_r \)  
Right legs angular velocity

\( r \)  
Wheel radius (Radius of leg path)

\( L \)  
Leg to center of miniROaCH distance

\( \alpha \)  
World to miniROaCH angle in Polar Coordinates

\( \rho \)  
World to miniROaCH distance in Polar Coordinates

\( v_u \)  
Radial velocity of miniROaCH

\( v_w \)  
Tangential velocity of miniROaCH

\( a_u \)  
Radial acceleration of miniROaCH

\( a_w \)  
Tangential acceleration of miniROaCH

\( I_m \)  
Armature motor current

\( R_m \)  
Motor’s internal resistance

\( L_m \)  
Motor inductance

\( E \)  
Motor’s back-emf

\( \tau_m \)  
Motor torque

\( k_t \)  
Motor’s torque constant

\( k_e \)  
Electromagnetic force constant

\( R_s \)  
Current sense resistance

\( \theta_m \)  
Motor’s angular displacement
Glossary

\( w_m \)  
Motor’s angular velocity

\( w_{\text{leg}} \)  
Angular velocity of the legs

\( J \)  
Moment of inertial of the motor

\( b \)  
Motor viscous friction constant

\( m \)  
Mass of the robot

\( J_r \)  
Inertia of the robot

\( d \)  
Distance between robot’s center and the center of mass

\( F_{uRi} \)  
Longitudinal force from miniROaCH’s ith right leg

\( F_{wRi} \)  
Lateral force from miniROaCH’s ith right leg

\( F_{uLi} \)  
Longitudinal force from miniROaCH’s ith left leg

\( F_{wLi} \)  
Lateral force from miniROaCH’s ith left leg

\( F_{uRtot} \)  
Total longitudinal force from miniROaCH’s right legs

\( F_{wRtot} \)  
Total lateral force from miniROaCH’s right legs

\( F_{uLtot} \)  
Total longitudinal force from miniROaCH’s left legs

\( F_{wLtot} \)  
Total lateral force from miniROaCH’s left legs
Chapter 1

Introduction

1.1 Motivation

The research conducted within this thesis found motivation from a few different sources. Primarily, the need for robust, affordable, and highly capable mobile sensors is a motivational factor which pushed the development of the miniROaCH robot. Over the years, research institutions and industrial companies have worked to develop and produce dozens, if not hundreds, of different robots of various capabilities and purposes [4]. Most mobile robotic platforms follow a semi-standard and common design when it comes to their form of locomotion. This common form of movement is wheeled motion [5], [6], [7], [8]. Wheels provide a simple and effective means of movement. These types of robots are well studied and the robotic market is arguably over-saturated with these particular types of robots. Their kinematics and dynamics are well understood, making them easily controlled. However, wheeled robots have certain limitations and drawbacks.

In recent years, researchers and scientists have been making many advances in biologically inspired robots. Robots which utilized legged motion have the potential
Chapter 1. Introduction

to navigate terrains which wheeled robots cannot. By taking a look at the gaits and locomotion mechanisms which evolution has created, advancements in mobile robots can be made at a much quicker rate. Evolution has spent millions of years performing locomotion on many scales; it is therefore smart and useful to learn from the organisms such a process has created. These ideas motivate the use of the miniROaCH six-legged platform.

Crawling robots have the potential of exploring environments and terrains which wheeled robots may be unable to. These environments could include collapsed building filled with rubble, uneven and rocky terrains, and even through small gaps under doors or fallen obstacles. Such scenarios could include search and rescue missions after an earthquake or a tornado. Sending humans into these environments are typically very dangerous due to structural uncertainty and the hazardous nature of a collapsed building. Cheap mobile sensors, or robots, can be used to explore and map these environments before sending in humans. With a swarm of affordable and disposable sensors, the survival of the individual agents is less important due to their cheap form factor and the ability to replace them if needed. This is a quality which most commercially available robots, and even those used for research, do not typically posses.

1.2 Problem Statement

The main objective of this thesis is to demonstrate the steps needed to transform a commercially available six-legged connected toy robot into a fully functioning and highly capable research platform which utilizes a set framework for robot coordination and control. To be utilized as a mobile sensor, the robot must be low cost and affordable. This is accomplished by incorporating readily available low-cost electronics for the various components needed to have a robot which is self-controlled
Chapter 1. Introduction

and fully autonomous using minimal external resources. The robot must not rely on external sensors such as a motion capture system or GPS.

Certain steps must be taken to achieve this goal on a unique legged robot. Although the robot may seem simple, the system is complex and certain aspects create challenging problems. It is difficult to derive an accurate model of the system dynamics which is needed for control; however, with appropriate approximations, it is possible to derive a model which can be effective and useful. The models used have been derived and explained. Low level motor control must be able to control the linear and angular velocities of the robot. Additionally, a high level position controller must show that the platform is capable of being controlled reliably. The robot must also be able to estimate its position and orientation within an environment using only on-board sensors and external landmarks within the environment. Lastly, the system must be used within a few applications to demonstrates its capabilities.

1.3 Related Work

1.3.1 Crawling Hexapods

Biologically inspired robots can be designed by observing and analyzing the key principles found in nature. In nature, locomotion for most animals is achieved via the use of legs that can range in quantity, shape, and size. Limb movements are triggered by muscles which create linear motion based on the structure of the animal. Nature consists of a vast majority of crawlers that roam the earth with extreme agility, especially when faced with obstacles. The common cockroach has been studied many times. Like most insects, the cockroach has three pairs of jointed legs. This type of body composition allows the cockroach to maintain stability during locomotion even at high speeds [9].
Chapter 1. Introduction

The cockroach’s locomotion has been studied for over 45 years [10]. [10] studies the locomotion of the Periplaneta Americana cockroach. This study and research was conducting to develop a better understanding of leg movement co-ordination and the mechanisms of control which a cockroach uses when in motion. According to this article, a cockroach utilizes an alternating tripod for almost all speeds. An alternating tripod (alternating triangle gait) is essentially one in which 3 legs are planted on the ground at any given time. These legs alternate repeatedly to propel the cockroach forward. At low speeds, this movement has a slightly different response. The results indicated by this research were concluded by physical experiments in which cameras were used to witness cockroaches running through a testbed which was set up to monitor the locomotion movements while running in a straight line. According to the results, the alternating triangle gate was almost never synchronous. In other words, the 3 legs of the alternating tripod did not leave the floor at the exact same time; however, the stepping pattern remained consistent for all stepping frequencies above 3Hz.

For the purposes of robotics, a cockroach can be classified as a hexapod. Hexapod locomotion is well studied and many papers can be found pertaining to this type of movement. [11] discusses the stability of hexapods in locomotion. Research has shown that at high speeds, hexapods such as cockroaches are not statically stable. To maintain stability at high speeds, cockroaches must have dynamic stability. Just as [10] set up an experimental testbed to monitor a cockroaches leg movement while walking/running, [11] presents a method for monitoring the dynamic stability in such instances. The experiment consisted of adding a perturbation while the hexapod (cockroach) was in linear motion to monitor the stabilization mechanisms used. High speed video was recorded of the locomotion and immediately following the testing, each cockroach specimen was deep frozen. Next, the center of mass and moment of inertia of the cockroach was determined. This data was used to determine how the cockroach is able to stabilize dynamically when a perturbation is presented. The
Chapter 1. Introduction

cockroaches were able to recover very rapidly and was determined to be possible due to properties of the musculoskeletal system which is able to stabilize using reflex. Such a study is useful when designing a biologically inspired hexapod.

Complex mechanical designs which accurately mimic a hexapod’s leg movement have been created for research purposes [12]. This article covers a design modeled off of the same cockroach studied in [10], Periplaneta Americana. The purpose of legged robots is to allow locomotion to be achieved in terrains in which traditional wheeled robots are unable to scale. Such terrains include rocky, muddy, and bumpy environments in general. Using a 6 legged agile insect such as the cockroach as inspiration will allow a robotic swarm of these biologically inspired insect-like robotic vehicles to effectively and efficiently achieve tasks in unknown and potentially dangerous environments.

In a robotics research setting, much work has been done at the Biomimetic Milisystems Lab at the University of California, Berkeley [13], [14], [15], [16], [17], [18]. The work in [13] presents a small and lightweight 16 gram hexapod robot which is able to move at speeds of 15 times its body length every second. Even more impressively, a 2.4 gram hexapod of similar locamotion is presented in [15]. [14] explores the dynamic stability of yet another 6 legged crawling robot, the VeicRoACH. With the help of design adjustments geared towards increasing running speeds, VeicRoACH is capable of speeds of 27 times its body length every second. This paper shows how the rotational dynamics have a large influence on a robot with the VeicRoACH’s size. Using their gait tuning methodology, regions of increased stability were found and were utilized in operation. Additionally, an aerodynamic roll damper was added to the system to further increase stability; however, this had an effect on maximum obtainable velocity.
Chapter 1. Introduction

1.3.2 Related Testbeds

Much work has been done previously to create multi-robot testbeds. [5] presents a Micro Multi-Vehicle Platform, or microMVP. In many ways, microMVP is similar to the miniROaCH testbed. However, there are some distinguishing differences. Like the miniROaCH testbed, the Micro Multi-Vehicle Platform was designed to be affordable, portable, and open source at the micro-scale. The vehicle is meant to be useful in both a single or multiple robot environment and is capable of performing centralized multi-robot path planning algorithms and distributed reciprocal velocity obstacle algorithms while maintaining a high accuracy of both state estimation and controller capabilities. MicroMVP is composed of the micro-vehicles, a state tracking camera system similar to that of the tabletop version of the testbed described in the Applications Chapter and a software stack that allows vehicle identification and control. The microMVP vehicle is composed of an Arduino micro-controller, Zigbee communication unit, two micro gear motors, a 3D printed plastic shell weighting less than 100 grams, and a power battery rated at 400mAh. Since the microMVP vehicle uses a simple micro-controller as the on-board controller, miniROaCH’s single-core ARM processor delivers many benefits in comparison. One convenient feature of the miniROaCH is the ability to remotely log-in to the robot through SSH and edit code on the fly. With an on-board ARM processor, the miniROaCH is able to handle an on-board 8MP video camera which allows for situational awareness and mapping capabilities. By incorporating a 9-axis inertial measurement unit, direct motor feedback, and fiducial landmark recognition for visual odometry, the miniROaCH is capable of traversing an environment in an autonomous manner without relying on external sensors.

Pickem et al. presents the Robotarium multi-robotic platform in [6]. Robotarium was developed to be a remotely accessible multi-robot laboratory. The robots designed for Robotarium are called GRITSBots. GRITSBots, like miniROaCHes, are
Chapter 1. Introduction

low cost, small differentially driven robots. However, like most multi-robot testbeds, [5],[6], [7], [8], the robots utilize wheels for movement. This is one advantage the miniROaCH possesses. Crawling robots are more capable on inclined surfaces and can handle uneven terrain. The e-puck robot presented in [7] hosts an array of sensors including proximity sensors, a microphone array, a 3D accelerometer, and a color CMOS camera. It is important to note, however, that the capabilities of the camera on the e-puck is severely limited by the dsPIC micro-controller used on the robot. In example, the e-puck is capable of grabbing a 40 by 40 pixel image off of the full image provided by the camera. The miniROaCH is capable of handling a full 8MP image. A more drastic difference between an e-puck and a miniROaCH is cost. An e-puck robot can cost nearly $1000 while a single miniROaCH can cost less than $200.

Although much work has been done to create these cost efficient and readily accessible multi-vehicle platforms, the miniROaCH testbed expands the capabilities and abilities of a single multi-robot testbed by providing low cost biologically inspired ground agents equipped a high definition color camera and an on-board single-core ARM processor at a cost that makes the robot accessible to most research institutions. Furthermore, the full miniROaCH testbed takes the multi-robot system further than a laboratory table by making the platform truly portable by introducing the heterogeneous aspect: the addition of an aerial vehicle for ground-agent position tracking as seen in one of the miniROaCH applications. Heterogeneous systems have been shown to improve mission efficiency [19]. West and Fierro explore the benefits of a hybrid (optical and radio) communication network with minROaCH ground agents and an overhead quadrotor in [20]. Cruz et al. also explores this subject in [21] by utilizing an aerial vehicle as a data mule to collect data from a ground agent via an optical communication link.
Chapter 1. Introduction

1.3.3 Robotic Exploration

Robotic exploration is a heavily pursued application. Typically, humans tend to search for technologies to aide with tasks which can be difficult, dangerous, or time consuming for people do. Therefore, using robots to explore unknown and possibly dangerous environments is an area within robotics that seems to receive a lot of attention. It is often useful to utilize a heterogeneous combination of robots to more efficiently explore an environment [22]. The work of [23] shows how to scale exploration with a large number of agents in which over 80 various robotic agents of different capabilities were used to explore and map an indoor environment.

1.4 Contributions

The main contributions of this thesis are

- miniROaCH: The miniROaCH robot which was developed for this thesis was described in detail on the hardware level. System components were described and discussed.

- Software: The software component of the miniROaCH platform is discussed. This includes system algorithms used within the Robotic Operating System, ROS.

- System Architecture: The system’s architecture is described. This includes the low-level and high-level controllers utilized. Additionally, state estimation is described and results are discussed.

- Position Control: Position control was implemented on the miniROaCH platform. Results are shown and discussed.
Chapter 1. Introduction

• MAST Demo: The miniROaCH robot was used for the Micro Autonomous Systems and Technologies, MAST, project as the ground agent platform which used the robot’s camera images and mobility capabilities to gather visual data to recreate a 3-dimensional map of the explored environment.

1.5 Organization of Thesis

Chapter 2 discusses the hardware and software used within this thesis which includes the details and descriptions of the hardware components and sensors used to create the miniROaCH robot. Additionally, the motion capture system used within the MARHES laboratory is presented. Robotic Operating System is discussed. In Chapter 3, the system’s kinematic and dynamic models are derived and presented. System assumptions are discussed. The architecture of the system from a high level is presented. In the next chapter, Chapter 4, the low-level and high-level controllers are shown along with the method of localization. The controllers and estimators are shown in implementation and the results are presented. A few applications which utilize the miniROaCH are discussed in Chapter 5. In the final chapter, Chapter 6, the conclusion of this thesis is explained. Areas of Improvement and future work are considered.
Chapter 2

System Overview

2.1 Hardware

2.1.1 miniROaCH

The miniROaCH, miniature ROs-enabled and Crawling Hexapod, is a small and light-weight biologically inspired, six-legged crawling robot. MiniROaCH takes inspiration from the cockroach. With three legs on each side of its body, miniROaCH mimics a cockroach’s locomotion mechanisms by implementing an alternating tripod gait as seen in nature. This means of movement is common in insects as it allows them to scale non-horizontal surfaces with ease. This fact can be exploited by miniROaCH robots, as the robots are able to navigate uneven terrains. The miniROaCH is therefore an excellent platform for robot exploration and mapping applications. The current version of the robot (see Figure 2.1) weighs approximately 70 grams. By utilizing off-the-shelf and readily available electronics, a single robot can be built for less than $200. Therefore, the miniROaCH is a cost effective solution to multi-robot and swarm research.
Chapter 2. System Overview

2.1.2 Physical Build

The miniROaCH chassis is an origami style body sold by Dash Robotics, Inc. [24]. The original Dash robot, Kamigami, was sold as a kit for $99 and was available for purchase online through Dash Robotics’ website. After partnering with a larger toy manufacturing company, Dash Robotics has redesigned the Kamigami robot and now sells the toy through large retail chains such as Walmart and Amazon for $49. Each kit consists of one full body chassis, a 1S battery, a two DC motor gearbox, and an electronic board used to control the robot via Bluetooth. However, the electronic PCB supplied with the kit is not necessary nor is it utilized on the miniROaCH. Instead, miniROaCH uses a Raspberry Pi Zero W as its processing unit. A custom 3D printable mount was designed specifically to house all of the miniROaCH electronics.
It is important to note that an effort was made to minimize the overall weight of the system. The original version of miniROaCH weighed nearly 40 grams heavier than the final version which caused issues with the motors and gearbox. The small DC motors were unable to create enough torque to move the robot with such a heavy load and as a result pulled too much current which caused damage to the motors. Revision 1 of the miniROaCH weighed approximately 120 grams. The third revision weighed less than 70 grams. However, this version was upgraded to house a variety of new sensors to allow for autonomous control. The current miniROaCH, revision 4, weighs 80 grams.

The original toy (version 1) Kamigami is approximately 50 grams. The toy
includes a gyroscope, accelerometers, infrared sensors (38 kHz modulated) for communication, Bluetooth connectivity, and visible light sensors. For processing and control, the toy utilizes a Cortex M0 microcontroller. For power, Kamigami comes equipped with a 1S 5C 250 mAh Lithium Polymer battery with an integrated charger. The toy was developed for kids as a platform which children can purchase, build, learn, and play with. It is a nice toy for many reasons. Since children are able to build the robot on their own, it has the potential to be a stepping stone towards an interest in robotics and can promote thinking. Such a toy can also push students towards engineering and science. However, the toy is very limited when it comes to actually using the platform for science and research.

\subsection{Electronics}

The miniROaCH makes use of a $10$ Raspberry Pi Zero W computer (Figure 2.5). The Raspberry Pi Zero W is the newest variant of the Raspberry Pi Zero with built-in
802.11n wireless LAN and Bluetooth 4.0. More impressively, the Raspberry Pi Zero W incorporates a 1 GHz, ARM11 single-core CPU, 512 MB of RAM, an expandable memory microSD slot ($10 for an 8 GB microSD), 40 exposed GPIO pins, and a CSI (Camera Serial Interface) camera connector all on a 65 mm by 30 mm board.

MiniROaCH utilizes an 8 megapixel image sensor ($30) capable of capturing high definition video as the robot’s primary sensor. The robot is capable of handing additional sensors such as distance sensors, Inertial Measurement Units (IMU), and light sensors. This allows for user customization depending on application.

Currently, two INA219 current sensors are used for direct motor feedback (Figure 2.6). The INA219 from Texas Instruments is a bi-directional current and power monitoring IC. These sensors provide the amount of DC current being drawn by each motor with a resolution of ±0.8 mA. High side voltage feedback is also provided by the sensor. This includes the bus voltage and the shunt voltage. The DC current draw is determined by running the load in series with a shunt resistor with a low resistance ($R = 0.1\Omega$) and measuring the shunt voltage with an internal differential
programmable-gain amplifier (PGA) and Analog to Digital Converter (ADC). The IC contains a Power Register, Current Register, and Voltage Register. An I²C interface is used to read and set the registers on the chip. The voltage of the load (motor) can also be determined from these measured values. The INA219 breakout board is available from [25].

The miniROaCH robot uses a 9 Degree of Freedom BNO055 Inertial Measurement Unit (IMU). This sensor (Figure 2.7) was developed by Bosch and utilizes an accelerometer, a gyroscope, and a magnetometer. Furthermore, this sensor incorporates a Cotrex M0 microprocessor to filter and fuse all of the raw sensor data internally. Orientation data can be received as either Euler angles or Quaternions. Linear accelerations, rotation vectors, heading, gravity, and angular rates are provided from the sensor. The Raspberry Pi Zero W is able to communicate with the sensor through UART. The BNO055 breakout board was designed by Adafruit and is available on their website [26].
Chapter 2. System Overview

Motors are controlled via I²C communication. A Texas Instruments DRV8835 (Figure 2.8) dual motor driver H-Bridge IC is used to drive the motors with up to 1.5 A each at a maximum of 11.1 V. A breakout board for this device is available from [27]. The power system consists of a 1S 380 mAh Lithium Polymer (LiPo)

![Image](https://via.placeholder.com/150)

Figure 2.9: Pimoroni LiPo SHIM available from [2].

battery and a Pimoroni LiPo SHIM power management board (Figure 2.9). This board uses a Texas Instruments TPS61232 Step-Up Boost Converter which operates at 96% efficiency and delivers the Raspberry Pi Zero W with 5 V. LEDs on the board indicate when the board is being powered and when the battery reaches a low voltage. The low battery LED turns on when the LiPo battery falls beneath 3.4 V. The board automatically shuts down when the battery reaches a voltage of 3.0 V to protect the battery from damage. The board includes an enable pin (EN) which if pulled low to ground, will initiate the board shutdown. Charging the battery is done externally using a USB 5 V LiPo battery charger. Since the miniROaCH was designed such that the batteries can be swapped in and out of the robot, users can
Chapter 2. System Overview

charge spare batteries and replace them with ease. Excluding the cost and time for

3D printing the body mount, a single miniROaCH can be assembled in less than half an hour for right around $200.

2.1.4 Communication Network

The miniROaCH testbed utilizes 802.11n wireless LAN for communication. Each miniROaCH is WiFi and Bluetooth 4.0 enabled. A wireless WiFi router allows each member of the testbed to communicate. Since each miniROaCH has an on-board processor running the Rasbian operating system, users are able to remote access the operating system through a Secure Shell protocol, or SSH. This allows users to directly monitor on-board processes. For example, a user can directly stream video
from a miniROaCH to a base computer. Individual miniROaCH robots are also able to communicate with one another through WiFi since they are all connected to the same access point. This is possible using Rosbridge [28].

2.1.5 Vicon System

The MARHES laboratory utilizes a Vicon motion capture system for many projects and demos. The Vicon system is a high accuracy object tracking system capable of providing position and orientation feedback with sub-millimeter (<0.5 mm error) and sub-degree (<0.5 degree error) accuracy. Within the laboratory, eight (8) Bonita mocap cameras are mounted within a netted arena which provides safety when testing with robots which posses a threat to safety such as aerial vehicles. The cameras shine bright red (680 nm) LEDs into the area. Objects fitted with fixed retro-reflective markers within the area reflect the light back to the eight cameras. The cameras are all connected to an MX Giganet which collects the video data which is then sent to a Windows 7 PC running the Tracker software. The Tracker software is able to provide position and rotation data at up to 225Hz. The cameras constantly update at a rate of 240 frames per second (fps).

Within the Tracker software, users are able to create an object by selecting the retro-reflective markers attached to the object. Once all markers on the object are selected, the user is able to give the object a unique name. Tracker then publishes the object information over the network to any machine running the ViconStream code provided by Vicon. In the case of the MARHES lab, a dedicated MAC Mini running Ubuntu 14.04 and ROS Indigo collects the data for redistribution. This is done via a ROS Node called mantiss_vicon. This node broadcasts a TF, or transformation) containing the object’s position and orientation data with respect to the world frame of reference. Another node, ros_bridge, runs in parallel on the same MAC Mini to send the topics published by mantiss_vicon over the network to any client machines.
Figure 2.11: Kamigami robot with retro-reflective markers within the Vicon motion capture area in the MARHES Laboratory.

requesting the information.

2.2 Software

2.2.1 ROS

Robotic Operating System (ROS) is an open source framework developed for writing robotic software. ROS was designed to be flexible and modular. The Robotic Operating System environment contains various tools and libraries which make integration, testing, and debugging software quite simple. Additionally, ROS allows
Chapter 2. System Overview

for complete customization while keeping packages and libraries consistent for users by setting in place standard ROS conventions. The software environment allows for hardware abstraction and handles internal communication between system processes.

ROS can be used for all aspects of robotics from sensor integration to high level system control. The core components of ROS include

- Communications Infrastructure
- Robot Specific Features
- Tools

The Communications Infrastructure which ROS provides handles many features such as anonymous and asynchronous message passing, recording and playback of messages, remote procedure calls, and a distributed parameter system. Message passing may be one of ROS’ most useful features. Individual processes, or Nodes, running within the operating system are able to communicate easily with each other through the anonymous publish/subscribe architecture. System structure is broken down into a node distribution. A node is defined in ROS as a process which performs calculations and computations. Nodes are distributed within a graph. Nodes communicate through the graph over topics, services, and the Parameter Server. Users are able to either use open-source and publicly available libraries which include provided nodes or develop their own to integrate within a system. Figure 2.12 shows an example graph of the node structure within ROS.

The simplest way for different nodes to communicate with one another is through a predefined ROS topic. Topics are streamed and broadcasted through the graph of nodes. A topic must conform to a certain predefined structure. The topic structure depends on the message type, i.e. std_msgs/String. The std_msgs/String message is defined by the std_msgs package and the message type is String. This message
Chapter 2. System Overview

Figure 2.12: ROS Node/Topic structure [3].

type is one of the simpliest, as the message contains a single variable, data, which holds a string. More complex structures exist. For example, the geometry_msg package contains more robotics specific types such as PoseStamped. A PoseStamped message structure contains two variables at the highest level: header and pose. The header variable is of std_msgs/header type and consists of a sequence identification number (uint32), a timestamp (time), and a frame ID (string). The pose variable is of geometry_msgs/pose type and consists of two variables: position and quaternion. Both position and quaternion contain three float64 variables named x, y, and z. The quaternion variable possesses a fourth float64 variable, w.

ROS topics are meant for one-way distributed communication with no built-in message request or reply functionality. ROS services handle this aspect of communication. A service consists of a request message and reply message. Just as topics are defined by message types, services are defined by service types.

The third node communication method is through the Parameter Server. The Parameter Server acts as a publicly accessible variable dictionary where different nodes can share variables and update their associated states. Private Parameter Servers can also be created and utilized.

Users can develop nodes in either C++ or Python. To implement C++ in ROS,
Chapter 2. System Overview

roscpp is used. For Python, rospy is used.

Packages

The following packages are used within this thesis in conjunction with custom nodes:

- roscpp - C++ integration with ROS
- rospy - Python integration with ROS
- std_msgs - Standard ROS Messages library
- cv_bridge - Computer Vision library
- geometry_msgs - Geometry ROS Messages library
- sensor_msgs - Sensor ROS Messages library
- image_transport - Image Transport library used for subscribing to and publishing images
- rosbag - Recording and playing back ROS topics
- gazebo - Integrated 3D physics simulator

Workspace Structure

To use ROS, users must utilize a catkin workspace. A catkin workspace is a directory in which users are able create and install existing catkin packages.

The catkin workspace on the miniROaCH is named "ros_catkin_ws". Within the workspace, a directory called "src" contains all of the packages needed to run the Robotic Operating System, i.e. "ros". This directory also contains all supporting...
packages used by the robot such as "raspicam" which allows the Raspberry Pi to integrate the camera with ROS. The directory tree in Figure 2.2.1 is shown as an example of the structure; however, it is important to note that this example does not contain all of the packages used on the robot. Over 35 packages were compiled from source on the miniROaCh within the catkin workspace to run ROS on the Pi Zero. Installing these packages from source and compiling the entire workspace on the Raspberry Pi Zero is a lengthy processes; therefore, the miniROaCH memory card has been cloned and will be available online. The "mast_pi_zero" directory shown contains all of the nodes described within this thesis. This code is online on
Chapter 2. System Overview

github [29].

Gazebo

The Gazebo Project is essentially an open source physics simulator designed specifically for roboticists. Gazebo is a highly capable physics engine which is able to model robots, sensors, and environments while providing realistic graphic rendering. Gazebo is well integrated with the ROS environment through the gazebo package. Because Gazebo is integrated with the ROS environment, users are able to rapidly and easily test algorithms, controllers, and AI systems within a realistic and accurate simulation before implementing on hardware. Due to their intertwined structure, ROS and Gazebo make the software to hardware transition seem seamless.
Chapter 3

Modeling and System Architecture

3.1 Kinematics Model

The inherent system kinematics of a miniROaCH robot can be modeled as an approximation of a differential drive mobile robot, or DDMR. Due to its method of locomotion, the miniROaCH robot can be classified as a nonholonomic mobile robot. This is done due to the fact that each side of the robot is driven by its own motor and compound gear train. When the left motor rotates, the left set of legs rotate in the same direction as the motor. The same applies for the right motor and right set of legs. Each leg rotates in a circular motion around the robot’s Y-Axis. The middle legs are 180 degrees out of phase in comparison to the outer legs.

A differentially driven robot can be controlled by changing the forward velocity and the angular velocity of the robot, \( v \) and \( \omega \) respectively [30]. To model such a system in state-space form, let the state matrix be \( \mathbf{q} \) where \( \mathbb{R}^3 \ni \mathbf{q} = [x \ y \ \psi]^T \). \( x \) and \( y \) corresponds to the position of the system in Cartesian coordinates and \( \psi \in [-\pi, \pi] \) is the heading of the robot. States \( \mathbf{q} \) are taken with respect to the world
Chapter 3. Modeling and System Architecture

frame of reference.

\[ \dot{x} = v \cos \psi \]  
\[ \dot{y} = v \sin \psi \]  
\[ \dot{\psi} = \omega \]  

However, in practice the model experiences process noise. The kinematic model of a differential drive robot can be extended based on the relationships between system geometries and the velocities of the robot. Given a DDR with two wheels of radius \( r \) and a wheel to wheel distance of \( 2L \), the kinematic equations can now be represented as

\[ \dot{x} = \frac{r}{2} (w_r + w_l) \cos \psi + \xi_1 \]  
\[ \dot{y} = \frac{r}{2} (w_r + w_l) \sin \psi + \xi_2 \]  
\[ \dot{\psi} = \frac{r}{2L} (w_r - w_l) + \xi_3. \]

It can be noted that \( \xi_1, \xi_2 \) and \( \xi_3 \) are the process noise parameters of the system. The process noise can be modeled as a Gaussian white noise. However, in simulations, the noise covariance is assumed to be zero. The algorithms tested in the simulations can be extended to take care of the noise characteristics accordingly.

Comparing the two models, it can be seen that

\[ v = \frac{r}{2} (w_r + w_l) \]  
\[ w = \frac{r}{2L} (w_r - w_l). \]

The states of the system are now dependent on two inputs, \( w_r \) and \( w_l \). Here, \( w_r \) and \( w_l \) correspond the the angular velocities of the right and left wheels, respectively.
Since the legs on the robot are driven in a circular motion by the DC motor similar to that of a wheel, the DDR kinematic model is used to approximate the kinematics of the miniROaCH platform.

It is common to use the kinematic model of a DDMR for closed loop control. For this it is assumed that the system is able to track velocity commands perfectly. [31] shows that if a low-level controller is able to track velocities at a rate much higher than that of the outer-loop position control than the kinematic model is sufficient. However, if the velocity loop does not perform quick enough with respect to the position loop, the dynamic model of the system is necessary.
3.2 Dynamic Model

To find the Dynamic Model of the miniROaCH, either the Lagrangian approach or the Newton-Euler approach can be used [30]. For this thesis, the Newton-Euler approach was used. Other methods have been used such as those in [32] where the dynamics were found by backstepping the kinematics. The following derivation is similar to that of a wheeled differential drive mobile robot. A free body diagram of the system is formed and the forces which are applied to the system are analyzed.

![Figure 3.2: miniROaCH dynamic model diagram.](image)

The inertial frame which the miniROaCH exists within is given by \( A \). The body frame \( B \) of the miniROaCH is positioned at the robot’s center of mass \( C \). The longitudinal and lateral velocity at the robot’s center of mass is given by \( \{v_u, v_w\} \). The
robot’s heading is given by $\psi$, and $\omega$ represents its angular velocity. The robot’s total mass is given by $m$ and the yaw moment of inertia with respect to the robot’s center of mass $C$ is given by $J$.

To begin, the body position and orientation in $\{W\}$ is defined using polar coordinates

$$\dot{\rho} = \rho e^{j\alpha} \quad (3.9)$$

Differentiating with respect to time to find the velocity and acceleration yields

$$\dot{\rho} = \dot{\rho} e^{j\alpha} + j \rho \dot{\alpha} e^{j\alpha} \quad (3.10)$$

$$\ddot{\rho} = \ddot{\rho} e^{j\alpha} + 2 j \dot{\rho} \dot{\alpha} e^{j\alpha} + j \rho \ddot{\alpha} e^{j\alpha} - \rho \dot{\alpha}^2 e^{j\alpha} \quad (3.11)$$

After grouping terms and simplifying

$$\dot{\rho} = [\dot{\rho}] e^{j\alpha} + [\rho \dot{\alpha}] e^{j(\alpha + \pi/2)} \quad (3.12)$$

$$\ddot{\rho} = [\ddot{\rho} - \rho \dot{\alpha}^2] e^{j\alpha} + [2 \dot{\rho} \dot{\alpha} + \rho \ddot{\alpha}] e^{j(\alpha + \pi/2)} \quad (3.13)$$

The radial and tangential velocity and acceleration terms are then

$$v_u = \dot{\rho}, \quad (3.14)$$

$$v_w = \rho \dot{\alpha}, \quad (3.15)$$

$$a_u = \ddot{\rho} - \rho \dot{\alpha}^2, \quad (3.16)$$

and

$$a_w = 2 \dot{\rho} \dot{\alpha} + \rho \ddot{\alpha}. \quad (3.17)$$

Therefore, the fundamental equations of acceleration of the miniROaCH are given by

$$a_u = v_u - v_w \dot{\alpha} \quad (3.18)$$
and

$$a_w = v_w - v_u \dot{\alpha}.$$  \hspace{1cm} (3.19)

Next, Newton’s Second Law is used to find the relationships between forces, torques, and accelerations.

The forces each leg produces is related to the torque created by the DC motors. This is the primary difference between a wheeled DDMR and a legged miniROaCH. A DC motor driven wheel maintains contact with the ground and produces a smooth forward and tangential force. Forces produced by the miniROaCH’s leg gait is discrete in nature, i.e. there are brief moments between gaits which produce negligible force on the ground. Deriving a true dynamic model of the miniROaCH which incorporates force discontinuity and the issues of including foot slippage/friction is out of the scope of this research. Instead, a few assumptions will be made.

First and foremost, it is assumed that the miniROaCH’s motors are able to provide a smooth torque function. This torque is assumed to translate to a smooth force distributed between the legs in contact with the ground. For example, there are a total of three legs per side. All legs are joined by an approximately rigid cross-member. Therefore, all legs on one side of the robot spin at the same rate. The two outer legs are in phase while the middle leg is 180 degrees out of phase. The legs have the same length and mass. With a constant torque over some time $\Delta t > T_m$ where $T_m$ is the time for a leg to complete one revolution, the forces between the two outer legs and the ground are assumed to be equal to the force between the middle leg and the ground on average.

$$F_{uLtot}(t) = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} \sum_{i=1}^{2} F_{uL_i}(t) dt = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} F_{uL3}(t) dt$$  \hspace{1cm} (3.20)

$$F_{uRtot}(t) = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} \sum_{i=1}^{2} F_{uR_i}(t) dt = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} F_{uR3}(t) dt$$  \hspace{1cm} (3.21)
Chapter 3. Modeling and System Architecture

\[
F_{\text{wLtot}}(t) = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} \sum_{i=1}^{2} F_{\text{wLi}}(t) dt = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} F_{\text{wL3}}(t) dt \quad (3.22)
\]

\[
F_{\text{wRtot}}(t) = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} \sum_{i=1}^{2} F_{\text{wRi}}(t) dt = \frac{1}{\Delta t} \int_{t_1}^{t_1 + \Delta t} F_{\text{wR3}}(t) dt \quad (3.23)
\]

The above equations are equivalent to saying

\[
\frac{1}{2} F_{\text{uL3}} = F_{\text{uL1}} = F_{\text{uL2}}. \quad (3.24)
\]

The Dynamic equations in their basic form are given by

\[
ma_u = F_{\text{uLtot}} + F_{\text{uRtot}}, \quad (3.25)
\]

\[
ma_w = F_{\text{wLtot}} + F_{\text{wRtot}}, \quad (3.26)
\]

and

\[
J_r \ddot{\psi} = (F_{\text{uRtot}} - F_{\text{uLtot}})L + (F_{\text{wRtot}} - F_{\text{wLtot}})d. \quad (3.27)
\]

Combining (3.18), (3.19), (3.25), (3.26), (3.27) produces

\[
u_u = v_w \dot{\alpha} + \frac{F_{\text{uLtot}} + F_{\text{uRtot}}}{m}, \quad (3.28)
\]

\[
u_w = -v_u \dot{\alpha} + \frac{F_{\text{wLtot}} - F_{\text{wRtot}}}{m}, \quad (3.29)
\]

and

\[
\ddot{\psi} = \frac{L}{J_r} (F_{\text{uRtot}} - F_{\text{uLtot}}) + \frac{d}{J_r} (F_{\text{wRtot}} - F_{\text{wLtot}}). \quad (3.30)
\]

It is important to note that the above equations are functions of \( \dot{\phi} \) which is defined in frame A. However, it can be shown that \( \dot{\phi} = \dot{\psi} \). Additionally, the tangential velocity of point A, \( v_w \), must be equivalent to \( d \dot{\psi} \) if there is no lateral slippage of the legs. This is another assumption made for this model. Substituting in \( v_w \) and
Chapter 3. Modeling and System Architecture

combining (3.28), (3.29), and (3.30) yields the non-holonomic constrained dynamic
equations of the miniROaCH robot

\[ \dot{v}_u = d\dot{\psi}^2 + \frac{F_{uL\text{tot}} + F_{uR\text{tot}}}{m} \]  \hspace{1cm} (3.31)

and

\[ \ddot{\psi} = \frac{L}{md^2 + J_r}(F_{uR\text{tot}} - F_{uL\text{tot}}) + \frac{mdv_u}{md^2 + J_r} \dot{\psi}. \]  \hspace{1cm} (3.32)

Further work must be done to relate the actuator dynamics to the forces created on
the legs.

3.2.1 Actuator Dynamics

A DC motor can be modeled by the following equation:

\[ V = I_m R_m + L_m \frac{dI_m}{dt} + E, \]  \hspace{1cm} (3.33)

where \( I \) corresponds to the armature current of the motor, \( R \) is the motor’s
internal resistance, \( L_m \) is the motor’s inductance, and \( E \) is the motors back-emf.
Chapter 3. Modeling and System Architecture

The armature current of the motor is proportional to the torque (3.34), \( \tau_m \), created by the motor and the motor’s torque constant, \( k_t \).

\[
\tau_m = k_t I_m \tag{3.34}
\]

This assumes that the strength of the magnetic field of the motor is constant.

Similarly, the back-emf, \( E \), of the motor is related to the angular velocity of the motor, \( \omega_m \), by the equation

\[
E = k_e \omega_m \tag{3.35}
\]

where \( k_e \) is the electromagnetic force constant.

The constant \( k_e \) was found experimentally by first measuring the internal resistance with an ohmmeter. Next, a known constant voltage was applied across the motor and the armature current was monitored. Since this experiment was conducted with no load applying any opposing torque on the motor, the current remained constant. A high speed camera recording video at 240 frames per second (fps) was used in post-process to calculate the speed of the motor in revolutions per minute (rpm). With a constant current, the term \( L_m \frac{dI_m}{dt} \) was ignored since \( \frac{dI_m}{dt} = 0 \). The electromagnetic force constant was found with

\[
k_e = \frac{1}{\omega_m} (V - I_m R_m). \tag{3.36}
\]

The gearbox which houses the 6 mm coreless DC motors provides a 16:1 reduction to decrease speed and increase torque. The legs are attached to the output shaft of the gear train in such a way that when the output shaft completes one revolution, so do the legs. The angular velocity of a leg is then equal to

\[
\omega_{leg} = \frac{1}{16 k_e} (V - I_m R_m - L_m \frac{dI_m}{dt}) \tag{3.37}
\]

Equation 3.37 is used to approximate the angular speed of the miniROaCH’s legs in implementation.
Equation 3.33 and the following equation derived using Newton’s 2nd law,

\[ J\ddot{\theta}_m + b\dot{\theta}_m = k_tI_m, \quad (3.38) \]

define the actuator dynamics of the system in which \( J \) is the moment of inertia of the motor and \( b \) is the motor viscous friction constant.

The individual motor dynamics in state space form is defined as,

\[
\frac{d}{dt}\begin{bmatrix}
\dot{\theta}_m \\
I_m
\end{bmatrix} = \begin{bmatrix}
-\frac{b}{J} & \frac{k_t}{J} \\
-\frac{K_e}{L_m} & -\frac{R}{L_m}
\end{bmatrix} \begin{bmatrix}
\dot{\theta}_m \\
I_m
\end{bmatrix} + \begin{bmatrix}
0 \\
\frac{1}{L_m}
\end{bmatrix} V, \quad (3.39)
\]

and the output equation

\[
\frac{d}{dt}\begin{bmatrix}
\dot{\theta}_m \\
I_m
\end{bmatrix} = \begin{bmatrix}
0 & 1
\end{bmatrix} \begin{bmatrix}
\dot{\theta}_m \\
I_m
\end{bmatrix}. \quad (3.40)
\]
3.3 System Architecture

The overall closed-loop system is defined by the system plant, a state estimator, and
the system controllers. Figure 3.5 shows the system structure in the form of a closed-
loop process control block diagram. The system utilizes a cascade control architecture
which consists of a low-level (LL) controller used to drive the miniROaCH’s legs to a
target angular velocity, and a high-level (HL) controller used to drive the miniROaCH
to a reference position.

Figure 3.5: Overview of the inner and outer closed loop system architecture imple-
mented on the miniROaCH.

In order for the response to kinematic based high-level controls to be sufficient
for the system, the low-level controller must handle the motor dynamics at a rate
much faster than that of the high-level loop [31]. In the case of the presented system
architecture, the inner-loop loop rate will be running faster than the high-level
loop rate. It is important to note that the miniROaCH platform was designed such
that other system architectures can be implemented in an easy manner by providing
a flexible software framework. For example, the system is also capable of receiving
feedback from external sensors such as a motion capture system. The high-level con-
Chapter 3. Modeling and System Architecture

trol loop was tested with feedback from the Vicon system in the MARHES testbed to demonstrate the possibilities of utilizing additional sensors for control. Additionally, an application is presented in which an downward-facing camera attached to an unmanned aerial vehicle (UAV) paired with 2D barcodes fixed to the miniROaCH can be used to create an interesting testbed that can be used virtually anywhere a quadrotor can fly and miniROaCH robots can crawl.
Chapter 4

Control Design, Localization, and Implementation

4.1 Controller Design

In order to control the miniROaCH on a high level, low-level control must be implemented to have reliable control of the two DC motors. Without the ability to set the leg speeds reliably with reasonable speed, a high level controller would be ineffective. DC motor control is a well established problem and has been studied for many years. Typically, the two dynamic motor equations which form the state space representation in Equation 3.39 are used along with a linear controller such as a PID or LQR to control the angular speed of the motor at the motor shaft, $w_m$. In most cases of motor control implementation the angular speed is available as system feedback provided by a sensor such as a rotary encoder. A rotary encoder could be either absolute or incremental. An incremental encoder could provide motion information of the shaft which can easily be used to calculate the speed of the shaft.

In the case of the miniROaCH system, there is no physical encoder which provides
the angular speed of the motor. Instead, current feedback $I_m(k)$ from the current sensors is used along with the applied voltage $V(k)$, the known armature resistance $R_m$, the experimentally found electromagnetic force constant $k_b$ and inductance $L_m$, and a first order numerical derivative (2 point derivative approximation) to approximate $\frac{dI_m}{dt}$ to calculate an approximation of the angular leg speed with Equation 3.37 with sample time $T_s$. This estimated angular leg speed, $\hat{\omega}_{leg}$, is then used as feedback to the controller,

$$\hat{\omega}_{leg} = \frac{1}{16k_e} (V(k) - I_m(k)R_m - L_m \frac{I_m(k) - I_m(k-1)}{T_s}).$$

This calculated $\hat{\omega}_{leg}$ is seen by the system as a state observer. Other forms of observers, such as a Luenberger Observer, could be used [33].

A Proportional Derivative (PD) controller was implemented on each motor. Figure 4.1 shows the low-level controller block diagram.

Figure 4.1: Low-level controller block diagram.

High level control of a robot is of high importance. Without the ability to command a robot to perform an action and receive a predictable response, a robot is more or less useless in many ways. This, of course, typically requires an accurate model of the system. As described in the Kinematic Models in Chapter 3, the miniROaCH has been treated as a non-holonomic differential drive wheeled robot under quite a few
assumptions. Of course, the largest assumption is that the system behaves similarly to that of a DDMR with wheels. The kinematic model of a DDMR relates the wheel speeds $\omega_r$ and $\omega_l$, the wheel radius $r$ and the length between the wheels $2L$ (or half of a leg step and distance between legs in this case), and the heading $\psi$ of the robot to the linear and angular velocity of the robot. Due to geometric properties of the miniROaCH body, the legs are treated as a wheel of radius 0.01 m positioned 0.1 m from the center of the robot. Figure 4.2 shows the block diagram of the high level controller implemented on the miniROaCH robot.

The heading_func() function takes in four arguments: $x_{\text{ref}}$, $y_{\text{ref}}$, $\hat{x}_{\text{ekf}}$, and $\hat{y}_{\text{ekf}}$. The output of this function is a reference heading, $\psi_{\text{ref}}$. Heading reference $\psi_{\text{ref}}$ is calculated off of basic trigonometric properties.

Figure 4.3 contains a diagram used to show how the reference heading, $\theta_{\text{ref}}$, is found. In terms of a mathematical function,

$$\psi_{\text{ref}} = \arctan\left(\frac{y_{\text{ref}} - \hat{y}_{\text{ekf}}}{x_{\text{ref}} - \hat{x}_{\text{ekf}}}\right).$$

(4.2)
4.2 State Estimator

For state estimation, an Extended Kalman Filter, or EKF, is used. An EKF is the non-linear version of a Kalman filter which linearizes the system around an estimate of the system’s mean and covariance. The use of Extended Kalman Filters is well established in control practice and acts as top choice for non-linear systems which are susceptible to Gaussian White Noise within both the process and measurement system. Feedback for the state observer is provided from two sources: a BNO055 IMU sensor, and two INA219 current and voltage sensors on the motors.

The first source of feedback is obtained from the BNO055 IMU through UART
communication. The gyroscope, accelerometer, and magnetometer within the IMU are already fused using an integrated microprocessor. Therefore, data fusion for the IMU sensors has already been handled and the data can be used directly as feedback for the observer. The data provided from the BNO055 IMU includes absolute orientation data in three-dimensional space provided in either Euler angles \((\theta, \phi, \psi)\) or Quaternions, angular rates \((\dot{\theta}, \dot{\phi}, \dot{\psi})\) in \(\text{rad s}^{-1}\), linear acceleration \((\ddot{x}, \ddot{y}, \ddot{z})\).

The second source of feedback comes from two INA219 current sensors. Each sensor is connected in series with a DC motor to read the current draw. The INA219 consists of a low resistance current sense resistor of 0.1Ω. Current can be read with an accuracy of 1%. The current and voltage are then input to Equation 3.37 with the known \(R_m, L_m,\) and \(k_e\) values. \(\frac{df_m}{dt}\) is determined using a two point derivative approximation as described in the previous section (see Equation 4.1. With angular leg speeds \(\hat{\omega}_r\) and \(\hat{\omega}_l\) (right and left angular leg speeds, respectively) and their mathematic relationships to the robot’s system’s linear and angular velocities, the miniROaCH is able to calculate a a kinematic-based estimate of \(\hat{x}, \hat{y}, \hat{z},\) and \(\hat{\psi}\).

\[
\hat{x} = \frac{r}{2}(w_r + w_l) \cos(\psi) \quad (4.3)
\]
\[
\hat{y} = \frac{r}{2}(w_r + w_l) \sin(\psi) \quad (4.4)
\]
\[
\hat{\psi} = \frac{r}{2L}(w_r - w_l) \quad (4.5)
\]

where
\[
\hat{x} = \hat{x} + dt\dot{\hat{x}} \quad (4.6)
\]
\[
\hat{y} = \hat{y} + dt\dot{\hat{y}} \quad (4.7)
\]
\[
\hat{\psi} = \hat{\psi} + dt\dot{\hat{\psi}}. \quad (4.8)
\]

The values out of the Extended Kalman Filter which are needed by the implemented system architecture are \(\hat{x}_{ekf}, \hat{y}_{ekf}\) and \(\hat{\psi}_{ekf}\).
Chapter 4. Control Design, Localization, and Implementation

4.3 Localization

The miniROaCH utilizes visual odometry as feedback for the kalman filter in estimating the position and orientation states of the robot. Visual odometry is the use of camera sensors to determine the position and orientation of the camera based on the associated images. The fiducial markers used in this research are called AprilTags, visual fiducial markers developed by the University of Michigan [34]. An AprilTag is a two-dimensional bar code encoded with a smaller amount of data than similar visual based systems such as QR codes. Each AprilTag can contain anywhere between 3 and 12 bits of data. By limiting the maximum data contained on each tag, accuracy, reliability, and range is increased of the visual tracking system. There are a total of four tag families contained in the AprilTag library. The Tag36h11 style family was chosen as it is recommended by the library’s developers.

![Figure 4.4: Tag36h11 style AprilTag.](image)

The visual odometry node runs at a much slower frequency than the motor feedback based odometry and the IMU based feedback. This is primarily due to the fact that the visual odometry is not done on the miniROaCH. Since the miniROaCH uses
Chapter 4. Control Design, Localization, and Implementation

a single core processor, the AprilTag library may be too computationally heavy to be run on the Raspberry Pi Zero W. Instead, the miniROaCH collects still images from its camera and sends the images to an off-board computer to do the heavy computer vision algorithms. The data transfer is done via Wi-Fi. Ideally, the images would be transferred at a faster rate via an optical wireless communication link. For the experiments conducted in this thesis, a Mac Mini with Ubuntu 14.04 and ROS Indigo was used to run the external visual odometry node. This node pulls the images from the miniROaCHes, determines its position and orientation in the world frame based off of the position and orientation of fiducial markers in the images.

ROS handles transformations between difference coordinate frames. This is done with ROS’ tf package. The tf package is able to handle multiple coordinate frames over time and provide a transformation tree structure. Essentially, the package performs cartesian coordinate transformations.

![TF tree structure](image)

Figure 4.5: TF (Transform) tree structure.

Figure 4.3 shows the TF tree structure captured while running a miniROaCH within the testbed. The testbed shown in Figure 4.8 contained five (5) AprilTags located at known positions within the environment. Each tag is unique. These fiducial tags act as known landmarks that the miniROaCH is able to use as the visual
odometry reference. Figure 4.9 shows the TF structure in 3 dimensional space. The five landmarks are connected directly to the origin of the ”map” frame while the miniROaCH is connected to one of the tags.

The TF tree in Figure 4.3 shows the ”map” frame connected to several children frames. Each direct child of the ”map” frame is a tag within the environment. These are provided by static transformations and are defined during setup. It can be seen that a transform is broadcasted by ”apriltag_detector” which connects ”camera” to ”tag_4”. A ”localize_tag” node running on an external computer listens for transforms from ”camera” to any of the eight (8) tags and re-broadcasts the transforms from the tag to ”camera_optical”. A static transform then connects the camera frame to the miniROaCH body.

Another node on the external computer, ”visual_odom_node”, listens for series of transforms which can connect the ”miniROaCH” frame to the ”map” frame. It then takes those transforms and finds the miniROaCH’s position and orientation with respect to the map coordinate frame and publishes it as a visual odometry message that the Extended Kalman Filter can use for localization.

### 4.4 Hardware Implementation

There are many challenges when attempting to implement all of the various system components in hardware. Each component must work efficiently and interface with the other components to achieve a powerful and highly capable platform. There are nodes used to control all of the individual hardware components which define the software architecture. Two forms of internal communication for the hardware components are utilized within the design. These two forms of communication are I²C and UART communication.
I²C communication is used to communicate with the two INA219 sensors and the motor driver board. The I²C communication protocol allows for a master device to communicate with multiple slave devices. In the miniROaCH hardware setup, the Raspberry Pi Zero W acts as the I²C master device. The two INA219 sensors act as two of the three I²C slave devices. For I²C, all slave devices must be connected in parallel with the master device. Four wires are used for I²C: Vcc, Gnd, SDA, and SCL. The Vcc line is used to power the slave devices and Gnd is used to set the circuit potential reference, i.e. a unified ground reference between all circuits/devices. SDA and SCL are the two serial communication lines for I²C. SDA is the serial data line while SCL acts as the clock line for the communication protocol. Each slave device has a unique I²C address identifier. In the case of the miniROaCH, the left motor INA219 has an address of 0x40 (INA219 default address) while the right motor INA219 has an address of 0x41. The right motor INA219 address was selected by shorting the A0 I²C address jumper on the INA219 board.

The third I²C slave in the system is the serial controlled motor driver. The I²C address for this device is 0x5F. A variety of commands are used to communicate with the motor driver. It is possible to enable and disable the motors, control the individual motors with forward and reverse commands, and invert the motors.

Communication with the BNO055 IMU is done through UART serial communication. Five wires are used to connect the BNO055 to the Raspberry Pi Zero W. These five lines are: Vcc, Gnd, TX, RX, and RST. Vcc is the 3.3 V voltage rail, Gnd is the Pi's ground reference, RX is the receive line, and TX is the transmit line.

4.5 Results

Figure 4.6 shows a plot of the linear (forward) velocity of the miniROaCH as a function of time given a step input with only the low-level controller running. The
Chapter 4. Control Design, Localization, and Implementation

Figure 4.6: Velocity Control Plot

robot was able to drive in a straight line for over 3 meters repeatedly. However, since the miniROaCH was driving with no position or orientation feedback, there were trials in which the robot would experience imperfections on the ground surface which induces a small hop on the robot causing it to change direction; however, the robot would remain at the constant linear velocity.

Figure 4.7 show the positional data of the miniROaCH collected from the Vicon motion capture system to show how the miniROaCH is able to crawl to a reference position, \((x_{\text{ref}}, y_{\text{ref}})\), while using the high-level controller described in the Controller Design section of this chapter. In both of the plots, the miniROaCH started at the same position of \((-0.4, -0.6)\) with a heading of \(-\pi/2\) rad. The first target position was \((-0.4, -1.9)\) while the second target position was \((0.4, -1.9)\). The miniROaCH was able to repeatedly achieve successful results. However, it is important to note that the circle regions on the plots show an area in which the miniROaCH accepts its position as a success. In other words, these circles define a success region for the robot. Once it enters the circle the miniROaCH registers this as a completed mission and continues on to the next way-point. Defining such regions is important
Figure 4.7: Position plots which shows the path a miniROaCH robot follows. The red circles represent a target region of success for the robot.

for the miniROaCH due to the fidelity of control capable on the platform. As stated before, the miniROaCH is limited by both weight and generated torque by the two 6mm brushed DC motors. Since the miniROaCH has a very course set of control, it is difficult for the platform to have fine tuned heading control. Therefore, the miniROaCH has a difficult time achieving precise movements. It is, however, able to reach areas of interest which is of higher importance in many scenarios.

Figure 4.8 shows a single miniROaCH in the testbed. The testbed is rectangular area with a width of approximately 1.5 meters and a length of approximately 2.3 meters. The testbed floor is made of corkboard. Corkboard was found to be a suitable material for the miniROaCH robot as it provides enough traction to minimize slipping but also smooth and continuous enough to allow the miniROaCH to crawl with its low-power motors. Figure 4.9 shows the corresponding position and orientation estimate provided to the miniROaCH from the ”visual_odom_node”. In Figure 4.8, the miniROaCH is facing the bottom right hand corner of the photo with the Apriltag
Figure 4.8: miniROaCH within the testbed environment. AprilTag are positioned within the environment as known landmarks to provide visual odometry.

Figure 4.9: TF connections which link the miniROaCH to a tag within the miniROaCH’s camera view and then to the map frame. The robot can be seen within the physical environment in Figure 4.8.

(tag4) furthest to the right within its camera’s view. The connections between the miniROaCH and tag4 can be seen in Figure 4.9 along with all of the connections
between the tags and the map coordinate frame. The miniROaCH receives these position and orientation updates at a rate of 2Hz whenever a tag is within the image frame.

Figure 4.10: Heading plot comparing ground truth measurements (Vicon), IMU data (Heading), and the output of the extended kalman filter (EKF).

Figure 4.11: Another heading plot comparing ground truth measurements (Vicon), IMU data (Heading), and the output of the extended kalman filter (EKF).
Figures 4.10 and 4.11 show the results from the extended kalman filter (EKF) which is providing an optimal estimate of the robot’s heading. The ground truth measurements are provided by the Vicon motion capture system. The values provided from the IMU are also shown for comparison. The results show that the EKF can provide a smooth and continuous estimate of the robot’s states for feedback. This is ideal as it acts as a filter which can reduce some of the measurement noise.
Chapter 5

Applications

5.1 MAST

5.1.1 Introduction

The research conducted in this thesis was motivated primarily by the US Army’s Micro Autonomous Systems and Technology (MAST) project. Micro Autonomous Systems can be utilized in many scenarios. They can provide support to certain missions which may contain challenges or dangers to humans. Scenarios such as search and rescue missions after natural disasters require an agent to navigate complex unknown environments which may be highly dangerous. Furthermore, many environments can be difficult to scale for the typically available robots. Such scenarios would require a novel approach to solving many of these difficult problems. The MAST project was created to tackle some of these issues by looking at robotics and technologies at the micro level.

Figure 5.1 shows an example cartoon in which a small swarm of crawling agents explore dangerous terrain while a group of Unmanned Aerial Vehicles (UAVs) gather...
Chapter 5. Applications

Figure 5.1: This figure demonstrates an example scenario in which a MAST system can be utilized.

information and data from the Unmanned Ground Vehicles (UGVs) to send to a cloud computer. This example motivated the research conducted in the MARHES laboratory in many ways. A few channels of research stemmed from this specific topic. For one, research was conducted into heterogeneity in wireless aerial and ground robotic networks. This resulted in the development of a hybrid communication network between aerial vehicles and ground vehicles via tradition radio-frequency (RF) communication methods such as 2.4 GHz Wi-Fi or Bluetooth and optical wireless
Chapter 5. Applications

(OW) communication means via lasers or infrared (IR) LEDs. The main premise is on the idea that a heterogeneous network utilizing both RF and OW would work more efficiently than one utilizing only one channel of communication. Optical wireless communication can be utilized for higher bandwidth communication while RF could be used for lower bandwidth status/data transfers between agents.

Figure 5.2: This figure demonstrates an example MAST scenario shown in the Vicon area within the MARHES Laboratory. Two quadrotors fly above a swarm of miniROaCHs and a Turtlebot. Four quadrotors can be seen on the floor on the right.

5.1.2 Architecture

Four types of agents existed within the MAST Demo Architecture. The first agent type was UAV. The UAV platform of choice for the MAST project was based on Astec Hummingbird quadrotors [35]. Each Hummingbird UAV was capable of waypoint
navigation [36] and relied on position and orientation feedback from the Vicon mo-cap system as described in Chapter 2.1. Each quadrotor was equipped with an optical wireless communication device, Wi-Fi capabilities, and an on-board Odroid XU4 computer which ran ROS. Since ROS was running on the aerial vehicles, interfacing with the second agent type, UGV, was simplified. The UGV platform was based on the miniROaCH. The miniROaCH architecture within a MAST context will be described later on in this section.

Figure 5.3: This figure demonstrates an example scenario in which a MAST system can be utilized.

The third agent type was a larger and more capable ground robot to act as the ground station. The Turtlebot was chosen as the ground station platform. The Turtlebot, like the other two agents, was running ROS for easy interfacing and control. This robot was equipped with a optical wireless communication device for high speed high bandwidth data transfers between itself and the aerial vehicles. This ground station was connected to the fourth agent type, ”the cloud”. For demonstration purposes, the cloud was a server within the lab. However, this could be replaced
Chapter 5. Applications

with a cloud server such as those provided by Amazon Web Services [37], [22].

5.1.3 Demo

To demonstrate the advances made for the MAST project, a demonstration was put together. The demonstration was a collaborative project done by myself and other members of the MARHES laboratory. The goal of the demo was to demonstrate the heterogeneous robotic group of robots utilizing a hybrid communication network. To do this, three (3) miniROaCHes were assembled as the mobile sensors/ground agents. The miniROaCHes' job was to explore an unknown environment and log video data locally using their 8 MP cameras. One or more Unmanned Aerial Vehicles could then fly above the individual miniROaCHes and initialize a video data transfer. Video data would then be transmitted from 'ROaCH to quadrotor via an optical wireless communication link. This would be preferred over an RF transfer due to the communication bandwidth necessary to transfer video data. The quadrotor would then act as a data mule and fly to another larger and more capable robot which could transfer the data to a cloud computer. In the case of this demo, a Turtlebot robot equipped with an IR optical transceiver and connected to a cloud server was used. Figure 5.2 shows a snapshot of the demo in which two quadrotors actively fly 'ROaCH to 'ROaCH collecting their video data and sending it to the Turtlebot robot which then sends the data to "the cloud". The server then processes the video data and position data of the mobile sensors and stitches together a map of the environment.

It is important to note that this work is presented to show the possibilities the miniROaCH robot presents. While all students helping within the project had their own particular roles, my focus was providing a robust and capable ground agent with a system architecture that could easily interface with the rest of the system, i.e. quadrotors, mapping, system control, etc.
Chapter 5. Applications

Figure 5.4: Generated 3D map overlay on an image of the actual environment.

Figure 5.5: Generated 3D map.

5.2 Heterogeneous Testbed with Aerial Localization

5.2.1 Ground Agent Tracking System

For a reliable swarm-capable system, accurate localization information is of high importance. The miniROaCH platform is capable of providing full state estimation for each ground agent with an overhead camera system. A single downward facing
Chapter 5. Applications

USB camera is utilized within this application of the system. A mvBlueFOX camera was chosen; however, since the system was designed to run on ROS, it is important to note that any standard USB camera with similar or greater performance specs will work sufficiently well. This application was done in conjunction with two other students from the MARHES laboratory. Once again, my role within the project was providing the physical hardware (i.e. the miniROaCH robot) that is able to handle the control structure needed for implementation; however, this is not to say that I did not play a large role in other aspects of the project. Much work was done in all system components such as setting up the software environment for the platform and handling all of the frame transformations in the system. Additionally, I played my part in development of the node architecture.

The miniROaCH platform allows for essentially two variant setups. The first setup is the tabletop setup. The tabletop setup provides users a simple and easy way to test their control algorithms and provides portability. The overhead camera is attached to a two-bar camera mount. The camera mount is attached to the table or desk with a 3D printable clamp. The camera housing and the bar joint is also 3D printable. 0.25 inch aluminum pipe was used for the two bars. The first bar was cut to 1m while the second was cut to 0.5m. As are all the files and build instructions, the STL and CAD files can be found on the MARHES website.

The camera mount is mounted on the center of the long edge of the table. This provides for maximum workspace. With the camera mounted 1 meter above the center of the table, a workspace of approximately 0.6m by 1.25m is available. Selecting a higher resolution camera for the tracking system should increase the size of the workspace, the accuracy, and the reliability of the tracking system.

The second setup is that of the heterogeneous nature. In this setup the tabletop camera mount is replaced with a mobile unmanned aerial vehicle, or UAV, equipped with a downward facing camera. This setup allows for the heterogeneous group to
Chapter 5. Applications

Figure 5.6: An example scenario with an overhead view of the tabletop testbed with 3 miniROaCH robots avoiding obstacles.

coordinate and collaborate. With this setup, the position and orientation of the UAV must be known. A VICON Motion Capture system was utilized for initial indoor testing; however, other methods and/or systems can be used to determine the pose of the UAV such as high precision GPS sensors for outdoor scenarios or visual odometry algorithms for indoor situations.

5.2.2 Pose Tracking

Both testbed setups rely on computer vision, image processing, and fiducial markers attached to each ground robot to determine the locations and orientations of the individual agents. With much thanks to the modularity which is inherent to ROS,
Chapter 5. Applications

Integrating image based state estimation is relatively straightforward. Each ground agent has a unique 10cm by 10cm AprilTag attached above its body. AprilTags can be described in Section 4.3.

The position and orientation of each miniROaCH is found through a series of coordinate frame transformations. The majority of these transforms are handled by ROS in a clean and efficient manner. The sequence of transformations that begin at the inertial frame (world frame of reference) and end at the camera is of high importance. Without the ability to transform the camera frame of reference to the inertial frame, positions and locations of the ground agents will be strictly relative to the quadroto’s (or fixed camera mount’s) frame.

Assuming the transform from the inertial reference frame to the body frame of the quadroto is handled (we used a VICON Motion Tracking System), another series of static transforms relating the quadroto’s body frame to the image frame of the camera will allow for the transforming between the inertial frame and the camera image frame. Since a unique AprilTag is rigidly attached to each miniROaCH, a static transform can be used to transform the miniROaCH’s body frame of reference to that of the AprilTag. At this point, the only remaining transform that would allow transformation between the inertial frame and the minROaCH’s body frame.
is the transformation matrix between the camera image frame and the AprilTag frame. This portion is handled by the AprilTag library which uses computer vision image processing algorithms to calculate the transform. Once the complete transform tree is built, the system is able to provide 6 DOF localization for each individual miniROaCH in the inertial frame.

The entirety of the localization transformation structure was implemented in ROS. ROS contains a TF library which is able to build the full transform tree within the system if all individual transforms are provided. As the miniROaCH testbed is a multi-agent platform at heart, it was important to develop the testbed in a modular way which allows for easy and seamless expansions. If a user would like to utilize another ground-agent, for example, the system must be robust and modular enough to allow this process to be quick and easy.

The miniROaCH testbed has a simple Node architecture. The testbed consists of three main types of nodes: an Operator Node, a Leader Node, and a Follower Node. The Operator node outputs the transform between the two input frames. This effectively provides the position and orientation of a robot with respect to a target frame. In example, to receive the position and orientation of a miniROaCH with respect to the world frame, the user would list the world frame as the leader frame id and list the miniROaCH robot’s base link frame id as the follower frame id. This Node was developed to run on either a base computer connected to an overhead camera for the tabletop version of the testbed or on the quadrotor’s on-board computer (an O-Droid XU4 in our case) for the full heterogeneous testbed. The Leader Node acts as the high level controller for the leader miniROaCH. The Follower Node acts as the high level controller for the follower miniROaCH robots. Modification of the Leader and Follower Nodes allows for an easy way for users to implement high level controllers. Both the Leader and Follower Nodes were developed to run on the miniROaCH robots for distributed algorithms or on base/quadrotor
Chapter 5. Applications

computer for centralized algorithms. For the centralized case, command velocities can be sent directly to each miniROaCH.

5.2.3 Gazebo Implementation

The full miniROaCH testbed has been developed within ROS [38] and Gazebo [39], a 3D robotics simulator. The quadrotor and miniROaCH ground agents were developed as modular ROS nodes and their associated C++ descriptions and classes. This allows for algorithms and code to be developed and tested within the simulated environment and then executed on hardware with minimal changes.

The default Gazebo world for the miniROaCH testbed is a flat and empty environment containing a single simulated AscTec Hummingbird quadrotor [40] and any number of miniROaCH ground robots. The Hummingbird quadrotor is equipped with a single downward facing camera located underneath the body beneath the center of mass. The Hummingbird quadrotor spawns at the origin of the Gazebo world. This can be changed by the user during setup if needed. The user is also able to specify the number of miniROaCH robots and their spawning locations and orientations.

Once the simulation has begun, the quadrotor executes a hover node which flies it to the altitude defined in the setup (default set to 1.5m) by the user. As the quadrotor reaches its hover point, the miniROaCH ground agents will enter the image frame of the downward facing camera. An increased quadrotor altitude will increase the viewable workspace; however, the further away the camera becomes from the ground agents the less effective the visual fiducial tracking system becomes. It is therefore important to find a balance between usable workspace and effective localization. At a height of 1.5m, and the camera parameters set to that of the mvBlueFox USB camera, the effective workspace was 1m by 1m. Within this space, three miniROaCHs were
Chapter 5. Applications

Figure 5.8: miniROaCH Virtual Testbed simulation in Gazebo (3 miniROaCH ground agents and one hovering Hummingbird aerial vehicle).

able to navigate with ample room. Figure 5.8 shows the simulated testbed in Gazebo with three miniROaCH robots beneath the Hummingbird UAV.

The first implementation in the virtual testbed consisted of three miniROaCH robots executing a leader-follower type controller, or follow-the-leader. In this simulated setup, one miniROaCH was chosen to be the leader of the pack. The job of the leader was to follow a predefined path within the workspace. This path was a circle centered at the origin of the Gazebo world frame with a radius of 0.4m. The quadro-
Figure 5.9: Top Plot: $x$ and $y$ position data w.r.t. the world reference frame of the Leader miniROaCH. Middle Plot: $x$ and $y$ position data of the first Follower miniROaCH w.r.t. the Leader miniROaCH’s reference frame. Bottom Plot: $x$ and $y$ position data w.r.t. the first Follower miniROaCH.

The leader robot was hovering at a height of 1.5m directly above the origin of the Gazebo world frame. Position and orientation data was provided to the leader by the overhead
quadrotor. This was done using an Operator Node which provided the transform from the world frame to the robot. A second robot was chosen to follow the circle-following robot. This miniROaCH used an Operator Node to receive a transform from the leader miniROaCH to itself. A Follower Node was then used by this robot to maintain a certain distance and heading (0.3 m and 20 degrees respectively) with respect to its target frame, that of its leader. This approach was used by the third miniROaCH, only its leader was the second miniROaCH. In other words, the first robot is leading the second and the second is leading the third. Snapshots of this virtual implementation can be seen in Figure 5.7. Results from this experiment can be seen in Figure 5.9.
Chapter 6

Conclusions

6.1 Conclusions

Within this thesis, a ROS-based framework was developed for a unique and powerful hexapod robot, the miniROaCH. The miniRoach robot is a flexible and versatile robotic platform which is able to crawl over obstacles, navigate potentially rough terrains, and even gather data to map environments. The robot is equipped with commercially available electronic components which makes the robot cost effective and easily reproduced. The current implemented framework and system architecture allows for users to easily adapt the system for certain research needs and can provide the foundation for swarm research.

Both the hardware and software components of the design have been presented and discussed. The platform is described on both a sub-system and system level. A kinematic model of the platform is shown. The dynamic model is derived and assumptions are explained. The chosen system architecture is presented and details regarding the low-level and high-level control architectures are given. Results from both the low-level and high-level controllers are shown, along with results from the
Chapter 6. Conclusions

state estimation implemented in the system.

Two applications are given in which the miniROaCH robot was used. The MAST project discussed shows how a heterogeneous group of robots which include miniROaCH robots as the ground agents of the system can work collaboratively for mapping and exploration by reconstructing data gathered by the miniROaCH robots into a 3 dimensional map of the environment. The second application, a Heterogeneous Testbed with Aerial Ground Agent Tracking, shows how an aerial vehicle equipped with a downward facing camera can provide the miniROaCH robots with localization information making the testbed expandable and truly mobile.

The miniROaCH platform incorporates many aspects of electrical and computer engineering including electronic design, visual processing, data processing, control theory, state estimation, mobile robotics, computer programming, and algorithm development. The resulting framework is extremely capable and is a great choice for research and development purposes.

6.2 Areas of Improvement

As with most experimental systems, there is room for expansion and areas of improvement. In particular, the leg’s angular speeds are estimated through an observer which is based on the actuator dynamics. However, motor encoders could be added to the system to receive direct measurements of leg speeds. This would result in more accurate control and odometry estimation. Additionally, the visual odometry implemented on the miniROaCH requires the use of an external processor for image processing. For the miniROaCH robot to be truly autonomous, all processing should be done by the robot. Therefore, it would be ideal to compile the visual odometry on the Raspberry Pi Zero W and test how the processor handles the computations. Lastly, integrating all of the electronics of the miniROaCh robot into a single printed
Chapter 6. Conclusions

circuit board (PCB) would reduce the total weight of the system, making it lighter, compact, and more robust.
Appendices

A  Circuit Schematics  1
B  PCB Layout  2
C  miniROaCH Revisions  3
D  CAD Model  4
E  Bill of Materials (BOM)  5
Appendix A

Circuit Schematics

The miniROaCH robot was originally designed to incorporate readily available commercial electronics to allow for easy modification of the Kamigami robot without the need to manufacture any custom components. However, the amount of electronics needed to have reliable on-board sensing and feedback is too demanding for the Kamigami motors used in this thesis. Therefore, cutting unnecessary weight is an important design aspect which must be addressed. The following circuit schematics show the circuit designed to replace a good majority of the off-the-shelf components with an all-in-one integrated electronic printed circuit board. This design incorporates the BNO055 9-DOF sensor, two INA219 current sensors, direct connections with the Raspberry Pi Zero W, and the dual H-Bridge motor driver with a supporting microcontroller.
Appendix A. Circuit Schematics
Appendix A. Circuit Schematics
Appendix A. Circuit Schematics
Appendix A. Circuit Schematics
Appendix B

PCB Layout

Figures B.1 and B.2 show the PCB layout used to replace the majority of the off-the-shelf components used on Revision E of the miniROaCH. This will cut unnecessary weight and allow users to assemble the robot in a fraction of the time. By using an integrated PCB as opposed to multiple boards, the system will have encounter less connection issues that can be caused by connecting the boards with wires. This will also allow for additional sensors to be interfaced with the robot.

Figure B.1: mRB: miniROaCH Board (PCB).

By integrating the majority of the off-the-shelf components, the cost of the platform can be reduced. This will make the platform an even more cost effective solution.
Appendix B. PCB Layout

for multi-robot/swarm research test-beds. Future revisions of the PCB can incorporate additional sensors to add functionality to the platform.

Figure B.2: mRB: miniROaCH Board (PCB) 3D view.
Appendix C

miniROaCH Revisions

Figure C.1 shows the original miniROaCH design. The original miniROaCH weighed nearly 120 grams. This version of the system left the original Kamigami PCB on the robot. This was unnecessary weight since the system did not utilize the PCB. Additionally, this system used one battery to drive the motors and another 2S battery to power Raspberry Pi Zero, the camera, a fixed output 5V DC-DC
Appendix C. miniROaCH Revisions

converter, and the motor driver control circuit. A USB Wi-Fi module was used for communicated with the robot.

The first miniROaCH revision, Figure C.2, was done in an attempt to cut the unnecessary weight from the original design. Heavy wires were replaced with smaller gauge and lighter wrapped wires. The Kamigami PCB was removed from the design. The Raspberry Pi Zero was replaced with the Raspberry Pi Zero W which had built in Wi-Fi and Bluetooth capabilities. This version was approximately 90 grams.

Figure C.3 shows the second revision. This version replaced the 2S battery and 5V step-down regulator used to power the Raspberry Pi Zero W and the electronics with a 1S battery and a small step-up voltage regulator, the same one used in the current revision addressed in the Hardware section of this thesis. The original motor driver was also replaced in this revision with the one used in this thesis. This version weighed approximately 82 grams.

Revision C (Figure C.4) cut weight by minimizing the amount of 3D printed material used.
Revision D (Figure C.5) was the first version of the miniROaCH to use a single battery to power both the motors and the electronics. Before Rev D, all batteries were fixed to the body with hot glue. Rev D redesigned the electronics mount to allow for a battery compartment underneath the Raspberry Pi Zero W which allows for
replacing batteries while doing experiments. This allows for users to have multiple charged batteries charged up and ready for experiments, as opposed to having to charge the robot between tests and experiments.
Appendix C. miniROaCH Revisions

The current version of the miniROaCH is shown in Figure C.5. This version is described in the System Overview Chapter within the Hardware Section.
Appendix D

CAD Model

Figure D.1: 3D CAD model rendering of the miniROaCH robot with the PCB, Pi Zero, and camera.
Appendix E

Bill of Materials (BOM)

<table>
<thead>
<tr>
<th>Item #</th>
<th>Name</th>
<th>Manufacturer</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Kamigami Robot</td>
<td>Dash Robotics</td>
<td>$49</td>
</tr>
<tr>
<td>2</td>
<td>Raspberry Pi Zero W</td>
<td>Raspberry Pi Foundation</td>
<td>$10</td>
</tr>
<tr>
<td>3</td>
<td>Pi 8MP Camera</td>
<td>Adafruit</td>
<td>$10</td>
</tr>
<tr>
<td>4</td>
<td>BNO055 IMU</td>
<td>Adafruit</td>
<td>$35</td>
</tr>
<tr>
<td>5</td>
<td>INA219 X2</td>
<td>Adafruit</td>
<td>$35</td>
</tr>
<tr>
<td>6</td>
<td>SCMD Motor Board</td>
<td>Sparkfun</td>
<td>$20</td>
</tr>
<tr>
<td>7</td>
<td>Pimoroni LiPo SHIM</td>
<td>Adafruit</td>
<td>$17</td>
</tr>
<tr>
<td>8</td>
<td>1S Battery</td>
<td>Tattu</td>
<td>$10</td>
</tr>
<tr>
<td>9</td>
<td>8GB MicroSD Memory Card</td>
<td>Adafruit</td>
<td>$3</td>
</tr>
</tbody>
</table>

Total Cost: $189

Table E.1: miniROaCH Bill of Materials.
References


References


References


References


