

# VISION-BASED TESTBEDS FOR CONTROL SYSTEM APPLICATIONS

by

ROBERT MICHAEL SIVILLI  
B.S.A.E. University of Central Florida, 2010

A thesis submitted in partial fulfillment of the requirements  
for the degree of Master of Science  
in the Department of Mechanical & Aerospace Engineering  
in the College of Engineering & Computer Science  
at the University of Central Florida  
Orlando, Florida

Fall Term  
2012

Major Professor: Yunjun Xu

© 2012 Robert Michael Sivilli

## **ABSTRACT**

In the field of control systems, testbeds are a pivotal step in the validation and improvement of new algorithms for different applications. They provide a safe, controlled environment typically having a significantly lower cost of failure than the final application. Vision systems provide nonintrusive methods of measurement that can be easily implemented for various setups and applications. This work presents methods for modeling, removing distortion, calibrating, and rectifying single and two camera systems, as well as, two very different applications of vision-based control system testbeds: deflection control of shape memory polymers and trajectory planning for mobile robots.

First, a testbed for the modeling and control of shape memory polymers (SMP) is designed. Red-green-blue (RGB) thresholding is used to assist in the webcam-based, 3D reconstruction of points of interest. A PID based controller is designed and shown to work with SMP samples, while state space models were identified from step input responses. Models were used to develop a linear quadratic regulator that is shown to work in simulation. Also, a simple to use graphical interface is designed for fast and simple testing of a series of samples.

Second a robot testbed is designed to test new trajectory planning algorithms. A template-based predictive search algorithm is investigated to process the images obtained through a low-cost webcam vision system, which is used to monitor the testbed environment. Also a user-friendly graphical interface is developed such that the functionalities of the webcam, robots, and optimizations are automated. The testbeds are used to demonstrate a wavefront-enhanced, B-spline augmented virtual motion camouflage algorithm for single or multiple robots to navigate through an obstacle dense and changing environment, while considering inter-vehicle conflicts,

obstacle avoidance, nonlinear dynamics, and different constraints. In addition, it is expected that this testbed can be used to test different vehicle motion planning and control algorithms.

To James and Nick for putting up with me

## ACKNOWLEDGMENTS

Every person I've met while at UCF helped to shape me into who I am today and I'm thankful for all of them, but there are a few people that deserve to be mentioned. I'd like to thank my advisor, Dr. Yunjun Xu, for his support, insight, and friendship. While we may not always have seen eye to eye (and occasionally butt heads), I'll forever be in your debt for everything you've done for me from GTA and GRA positions to helping to get my foot in the door at AFRL.

Dr. Jihua Gou and Dr. Hyoung Jin Cho: thank you for being on my thesis committee; your wisdom and guidance were essential.

Kristen Ryan: you are my love and my world. Thank you for being there for me while I worked through grad school.

Michael Sivilli and Karen Beck: for your love and support as my parents.

Dr. Randy Allen: you helped to spark my interest in hardware and application and so much more. You were a mentor and a friend and I wish you the best of luck in the future.

Dr. Khanh Pham: I cannot thank you enough for giving me the opportunity to be a part of the Space Scholars program. Also, thank you for teaching me to never eat something called "head-cheese" ever again.

Dr. Brien Flewelling: thank you for your insight, suggestions and friendship. While I may never be able to implement half of what we talked about, I always enjoyed the mental exercise of the discussion.

Dr. Gareth Basset: thank you for your work with bvmc and for all the help when we were first testing with the lego robots.

Fei Liang: thank you for your time, patience, and all of the material samples we used in testing. I'm sorry that we couldn't accomplish all that we hoped too, but I wish you the best of luck in the future.

Additionally, I would like to thank Brad Sease, Charles Remeikas, Jacob Belli, Puneet Vishwakarm, He Shen, Ni Li, and Moinul Forhad for all the times we had working together. You set the bar high as peers, and made great friends.

## TABLE OF CONTENTS

LIST OF FIGURES .....	xi
LIST OF TABLES .....	xiv
CHAPTER ONE: INTRODUCTION.....	1
Vision Systems.....	1
Shape Memory Polymers.....	1
Minimum-time Robot Obstacle Avoidance Path Planning.....	3
Testbeds .....	5
Thesis Outline .....	6
Contribution of the Thesis .....	7
CHAPTER TWO: WEBCAM-BASED VISION SYSTEMS .....	9
Camera Model.....	9
Compensating for Lens Distortion.....	9
Camera Calibration .....	12
Stereoscopic Reconstruction .....	15
CHAPTER THREE: VISION BASED CONTROL SYSTEM APPLICATION – SHAPE MEMORY POLYMER TESTBED .....	20
Point Tracking/Angle Characterization .....	20
Hardware Setup.....	22



PID Control .....	24
System Modeling .....	29
Identification .....	32
LQT Control.....	33
User-Interface .....	36
<b>CHAPTER FOUR: VISION BASED CONTROL SYSTEM APPLICATION- MOBILE ROBOT</b>	
<b>TESTBED .....</b>	<b>38</b>
BVMC.....	38
Hardware in the Testbed .....	39
Robot Platform.....	40
Robot Localization and Obstacle Detection .....	42
Path Tracking.....	45
Automatic Robot Parameter Calibration.....	46
Graphic User Interface .....	50
Wavefront Path Planning .....	53
Testbed Simulations-UCF.....	55
Tesbed Simulations –AFRL.....	60
Test Scenario 1 .....	60
Test Scenario 2 .....	61

Test Scenario 3 .....	62
CHAPTER FIVE: CONCLUSIONS AND FUTURE WORK.....	64
Conclusions.....	64
Future Work.....	65
SMP Testbed .....	65
Mobile Robot Testbed.....	65
LIST OF REFERENCES .....	67

## LIST OF FIGURES

Figure 1: Radial distortion contributions: <i>i</i> ) negative (barrel) <i>ii</i> ) no distortion <i>iii</i> ) positive (pincushion) .....	10
Figure 2: Chessboard used for calibration with inside corners (points tracked) highlighted in red. ....	14
Figure 3: Example of images used in calibration.....	14
Figure 4: Image from camera before (left) and after (right) distortion calibration.....	15
Figure 5: Perfectly aligned stereo setup.....	16
Figure 6: A general plot of disparity versus distance. ....	17
Figure 7: Sample with placed colored dots (left) and painted dots (right) .....	20
Figure 8: Reduced search area and associated dots .....	21
Figure 9: Two different examples of deflection tests tracking multiple dots. ( <i>left</i> )Dot locations and b-spline curve fitting. ( <i>right</i> )Linear curve fitting between dots.....	21
Figure 10: Example of sample test measure tip deflection angle (deflection angle) .....	22
Figure 11: Testbed for memory polymers .....	23
Figure 12: Data flow of the SMP testbed.....	23
Figure 13: Control response for ten different samples.....	26
Figure 14: Typical proportional gain input.....	27
Figure 15: Typical integral gain input .....	27
Figure 16: Typical derivative gain input.....	28
Figure 17: Typical power requirement of the PID controller .....	29

Figure 18: Responses of models compared to measured results for a) 22v input b) 24v input c) 25v input d) 26v input e) 28v input. ....	33
Figure 19: Deflection response of model to LQT controller ( <i>black</i> ) and five percent error bounds ( <i>red</i> ).....	35
Figure 20: Voltage input to model from LQT controller.....	35
Figure 21: User interface for SMP testbed .....	37
Figure 22: Testbed design for AFRL mobile testbed.....	40
Figure 23: Robot used in testing algorithms .....	41
Figure 24: Vision system identifiers used by robots (left and right) and example obstacle (middle).....	42
Figure 25: Visual output of a NCC using the obstacle template. a) After pass of NCC b) Original image c)after thresholding .....	43
Figure 26: Visual output of a NCC using the robot dot template before thresholding. Based off the same original image seen in Figure 25.a.....	44
Figure 27: Examples of path tracking ( <i>blue diamonds</i> ) for given trajectories ( <i>red</i> ) .....	46
Figure 28: Body and global reference frames used in testbed.....	48
Figure 29: Example of automatic calibration for a robot. Commands alternate between positive and negative rotations and translations. ....	49
Figure 30: Overall view of the GUI.....	50
Figure 31: Easy to use connection management of both the vision system and the robot connections .....	51
Figure 32: GUI showing obstacles detected and being divided into the two sections of the run .	52

Figure 33: GUI showing path planning and path tacking results of a typical run .....	52
Figure 34: Robot Viewer displaying raw sensor data, vision system information, and path generation info .....	53
Figure 35: Iteration 5(top) and iteration 6 (bottom) of an example wavefront path planning.....	54
Figure 36: Path found from wavefront path planning.....	54
Figure 37: The minimum time trajectory planned considering the known three obstacles in the first section (Case 1) .....	56
Figure 38: The minimum time trajectory re-planned considering all five obstacles in the second section (Case 1).....	56
Figure 39: Combination of the first and second sections of a testbed run (Case 1) .....	57
Figure 40: Combination of the first and second sections of a testbed run (Case 2) .....	58
Figure 41: Combination of the first and second sections of a testbed run (Case 3) .....	59
Figure 42: Test Scenario 1 ( <i>left</i> ) Section 1, ( <i>right</i> ) Section 2.....	61
Figure 43: Test Scenario 2 ( <i>left</i> ) Section 1, ( <i>right</i> ) Section 2.....	62
Figure 44: Test Scenario 3 ( <i>left</i> ) Section 1, ( <i>right</i> ) Section 2.....	63

## LIST OF TABLES

Table 1: Intrinsic and distortion camera parameters from calibration .....	15
Table 2: Measured and Calculated Values for the Example Calibration.....	49
Table 3: Testbed Result Case I .....	57
Table 4: Testbed Result Case 2.....	59
Table 5: Testbed Result Case 3.....	60

## **CHAPTER ONE: INTRODUCTION**

### Vision Systems

Since the 1980s, there has been a strong interest in the applications of vision systems[1, 2]. They offer a nonintrusive means of observing [3-5], measuring [6-8], and modeling [9-11] the surrounding world while advances in processing speed, lens production and imaging technology have opened doors for further applications. In the world today, they are seen everywhere from the dreaded red-light cameras at intersections[12], to video games and entertainment[13, 14], and in research labs everywhere [1-11]. This work presents two different applications of vision systems in control system applications.

### Shape Memory Polymers

Shape memory polymers (SMPs) are a kind of smart material which can be deformed and fixed into a temporary shape and later recover the original permanent shape with some external excitation such as heat, electricity, light etc. Recently, shape memory polymers (SMPs) have gained lots of attraction due to their outstanding properties. Compare to traditional shape memory alloy materials, SMPs possess many advantages such as easy fabrication, high elastic deformation capability and can have a wide range of activation temperatures compared to shape memory alloys [15-17]. SMPs have a wide range of applications, such as, but not limited to, textile and clothing manufacturing, biomedical engineering, and aerospace engineering [17-20]. As with many shape memory materials, direct temperature heating is the only actuation method of shape recovery. However, significant research has been devoted to finding various sources of this heating. Different kinds of actuation methods explored include infrared light heating[20],

laser light heating [18], electrical resistive heating[21], magnetic field [22, 23] and solvent[24-26].

Significant achievements have been made with SMP composites to date and shape recovery actuation can easily be achieved through the power dissipation of an electrical current through a conductive material. However, most previous work simply blended conductive filler into the pure polymer. This approach may lead to a number of problems if the polymer were used as matrix for fiber reinforced polymer composites and manufactured by resin transfer molding (RTM) processing. With this in mind, a unique technique of making conductive CNFs into paper form has been explored for functional materials [27, 28]. High-quality self-assembled CNF papers are produced by filtrating on hydrophilic or hydrophobic membrane. It's well known that CNF nanocomposites can significantly improve the mechanical performance of polymer, ceramic and teal matrixes with a small amount of CNFs.

The salient features of SMPs have made them promising in a growing number of applications. For examples, some SMPs can be used to develop biodegradable sutures, actuators, catheters, and stents for medical applications if made of the correct material [29, 30]; the SMPs are excellent materials for deployable and morphing structures of aircraft and spacecraft [31], ranging from array bearing hinges to thin-film structures [32] due to the material's light weight and malleability.

To make SMPs effective in the mentioned applications, precisely controlling the deflection is necessary. As seen in [33], SMPs can be controlled between their original shape and their current shape(in one direction) by regulating the material's temperature around  $T_g$  . This is because the memory effect stops when the SMP is below this temperature and will hold whatever



position it is in until it is reheated. Grouped with a system of precise control, this feature could be exploited to improve current and future applications of SMP. This precise control would give SMPs embedded actuators and tools a full range of possible poses dramatically thus increasing their practicality. Deployable structures made of SMPs could also be controlled to have desirable rates of deployment. This work delves into such precise control of SMPs using model based and model-less methods.

#### Minimum-time Robot Obstacle Avoidance Path Planning

Optimal path/trajectory planning for mobile robots is an important area of research that has attracted a lot of attention in recent years, and application of this research has expanded into fields of study such as handling of materials [34], exploration[35], transportation networks [36], and hospital care [37]. Many such systems, possibly with nonlinear dynamics and different constraints, are performed in complex and dynamic environments, where information such as obstacles may frequently change, and often mobile robots are expected to re-perform path /trajectory generation tasks quickly to account for new information and react accordingly.

Many methods have been proposed for mobile robot path/trajectory planning problems, ranging from classical approaches to heuristic approaches [38-41]. Some of these approaches are designed to find the optimal path within the search space. A formulation for efficient network flow is used in [42] to find optimal paths in arbitrary cost maps. As another method, a geometry maze routing algorithm[43] is used to obtain collision-free optimal paths within a workspace. In [44], Snell's law and optics laws are used to reduce the problem to an efficient graph search in order to find the optimal path for irregularly shaped regions that are homogenous. A significant downside of many optimal path planning methods is the significant runtime when considering

nonlinear dynamics and realistic constraints, which makes them difficult to implement in real time applications.

Other approaches are designed to find feasible paths rapidly. The analytical solution in [45] uses a family of piecewise polynomials to find feasible collision avoidance trajectories. Rapidly exploring Random Trees (RRTs) are used in [46] to find feasible paths while considering kinematic and dynamic constraints. A method involving incremental search on a multi-resolution and feasible lattice search space is used in [47] for fast planning and re-planning of feasible trajectories. These types of methods tend to find solutions quickly, but the feasible solutions are likely not equal or close to the optimal path.

In the method applied in Chapter 3, the focus is on finding an optimal trajectory solution while taking into account nonlinear dynamics, different state and control variable constraints, and obstacle/collision avoidance. Previous research has shown that the virtual motion camouflage (VMC) method, first introduced in [48, 49], can quickly solve for optimal solutions within a fixed or sequentially refined subspace formed by certain parameters, which in practice allows for a faster runtime, especially in complex environments. The B-spline augmented VMC (BVMC) method, introduced in [50], expands upon the VMC method by allowing the subspace to be optimized simultaneously, which can therefore achieve an optimal solution in the problem search space without significant increase in runtime. In addition, the method allows for more intuitive initial guesses, as the optimization parameters have physical meaning and can be easily guessed by the user.

### Testbeds

Prior to the application of any new algorithms to high cost and high risk real physical systems, it is reasonable to have an intermediate testbed. Testbeds are a controlled environment for the implementation of a theory or method to a physical system where the cost of failure, if any, is significantly reduced. A number of testbeds have been developed previously by other groups for different applications. For example, an unmanned air vehicle testbed is developed [51] using inexpensive and durable foam flying wings for further pursuing experimental research. In [52], a testbed supporting a ground based sensor network is designed using a larger number of overhead cameras and specialized robots and implemented as enhancement to limited simulations and models. The testbed seen in [53] physically constrains the rotorcraft being tested for added safety. Additionally, in [54] a large number of cameras are used in a testbed for sensing networked robot systems, with an emphasis on vision processing algorithms. Each of these was developed with a specific task in mind, with varying budgets and different workspaces, but all were developed with a common goal of achieving autonomous systems.

The testbed presented in this paper for shape memory polymers is designed to work with any variation of the SMP structure assuming a voltage driven actuation. Features of this testbed include: a low-cost, webcam-based vision system for measuring physical shape changes; current and temperature sensors for additional monitoring of SMP states; the capability to test various deflection control algorithms; and the capability to create linearized approximations of samples from test data.

The testbed presented in this paper for mobile robot systems are general enough to support the testing of new algorithms for both homogenous and heterogeneous autonomous

small-size vehicles, while the main focus is for ground-based systems in obstacle dense environments. Features of this testbed include: an overhead, low-cost, webcam-based vision system for robot localization, which is capable of reporting changes in the operating environment; a graphical user interface (GUI) that simplifies the management of robots and the settings of algorithm specific variables; a structurally changeable robot platform that permits the testing of heterogeneous systems; and a wireless connection on each robot and a central computer allows data to be shared as needed. The wavefront path planning algorithm[55], to find obstacle free corridors without considering robot dynamics, is used to enhanced a recently developed method, B-spline augmented virtual motion camouflage (BVMC) [50], which is tested in this testbed.

### Thesis Outline

Chapter two covers the fundamentals of the vision systems seen in both testbeds. The pinhole camera model will be introduced, how distortions found in the camera lens and imager can be compensated for, and a method of camera calibration will be presented. Additionally, the theory behind obtaining 3D position information about an object in the real world from a two camera system will be covered.

Chapter three presents the testbed designed for the testing, identification, and control of SMPs. A brief explanation of the particular type of SMP and the initial tests showing the macroscopic memory effect is initially covered. Then the method of identifying objects of interest for the vision system is covered followed by a description of the full hardware/control setup. An explanation of PID control is given and an example of deflection angle control is shown to work on a number of samples. Additionally, an ideal linearization is present and a

simulation model for sample response is created. A graphical user interface for fast and simple testing is also presented. This chapter is based off the published works [56-58].

Chapter four presents the mobile robot testbeds designed for the development of minimum-time and obstacle avoidance trajectory planning. A brief explanation of the BVMC method being tested is given followed by a description of the non-robot hardware implemented. The robot platforms used in testing are then covered, as are the means of locating them and detecting obstacles through the vision system using templates. A fast method of trajectory planning is implemented as a way of increasing the robustness and speed of the BVMC algorithm. Results of testing on both testbeds are then presented. This chapter is based off the published works [59-61].

In Chapter five, conclusions are drawn and suggestions for extensions of this work are given.

#### Contribution of the Thesis

The major contributions of this thesis are:

**Mobile Robot Testbed:** Two vision-based mobile robot testbeds were designed and constructed. One is housed in the guidance and navigations lab at the University of Central Florida, while the other is located at the space vehicles directorate of the Air Force Research Lab. Both were designed to be easy to use, support heterogeneous robot platforms, and assist in the development of autonomous, multi-body, cooperative, and optimal algorithms. In the development of these testbeds, the bio-inspired BVMC method of minimum time trajectory optimization was shown to work in a physical system and was further improved with the implementation of wavefront path planning which added to the convergence speed and

robustness of the algorithm in dense obstacle environments. This has laid ground work for the future development in guidance and control of robot/vehicle systems at the University of Central Florida.

**Shape Memory Polymer Testbed:** A vision-based testbed was designed, constructed and housed in the University of Central Florida's guidance and navigation lab. This testbed was used for the macro-scale system identification and precise control of the deflection angles of shape memory polymers, something that, to date, has not been done before. This will be implemented to further develop the control of SMPs for the foreseeable future.

## CHAPTER TWO: WEBCAM-BASED VISION SYSTEMS

### Camera Model

Given an object's pixel coordinate  $[u, v]$ , its global coordinate can be determined using the pinhole camera model [62] as

$$s[u, v, 1]^T = A[R | \mathbf{b}][X, Y, Z, 1]^T \quad (1)$$

where  $s$  is a normalizing scale factor,  $\mathbf{b}$  is the translation bias from the origin of the global coordinate system  $[X, Y, Z]$  to the camera centered coordinate system  $[x, y, z]$ ,  $R$  is the direct cosine matrix from the global coordinate system to the camera centered coordinate system with a rotation sequence of 3-2-1. Matrix  $A$  contains the camera's intrinsic parameters of focal lengths  $[f_x, f_y]$  and principal point locations  $[c_x, c_y]$  found through calibration [63] such that

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Equation (1) can be reduced to

$$s[u, v, 1]^T = A[x, y, 1]^T \quad (3)$$

without loss of generality via the assumption that  $Z = 0$  [64].

### Compensating for Lens Distortion

It is worth noting that  $[u, v]$  are the undistorted pixel coordinates based on the idealized model [63]. To account for the imperfections and irregularities found in many mass produced cameras, an expression to account for positional error must be considered to find the distorted, actual observed coordinates  $[u', v']$  such that

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} ud_r(u,v) + d_{ud}(u,v) \\ vd_r(u,v) + d_{vd}(u,v) \end{bmatrix} \quad (4)$$

Where  $d_r(u,v)$  and  $d_{id,i=u,v}(u,v)$  are the two major contributions of distortion in the model proposed by [65]: radial and tangential. Radial distortion mainly comes from flaws in the radial curvature curve of the camera. A high negative radial distortion creates a barrel effect where points become more crowded, thus decreasing the scaling between each point, as the distance from the optical center increases. Conversely, a high positive radial distortion separates points, thus increasing the scaling between them, creating a pincushion effect. Both instances can be seen in Figure 1

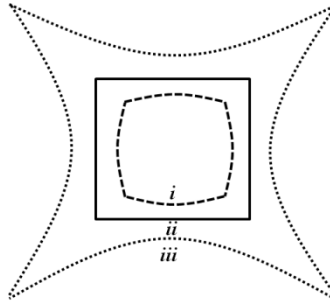


Figure 1: Radial distortion contributions: *i*) negative (barrel) *ii*) no distortion *iii*) positive (pincushion)

This distortion can be modeled as[66]

$$d_r(u,v) = 1 + k_1(u^2 + v^2) + k_2(u^2 + v^2)^2 + k_3(u^2 + v^2)^4 \quad (5)$$

where  $k_i, i = 1, 2, 3$  are coefficients of radial distortions to be found through calibration.

Tangential distortion accounts for random flaws in the lens (not related to curvature) and assembly mistakes and can be equated to [67]



$$d_{ud}(u, v) = 2p_1v + p_2((u^2 + v^2) + 2u^2) \quad (6)$$

and

$$d_{vd}(u, v) = p_1((u^2 + v^2) + 2v^2) + 2p_2u \quad (7)$$

where  $p_i, i = 1, 2$  are components to be found with calibration.

For the applications discussed in this work, a Newton-Raphson solver was written in C for finding  $[u, v]$  given the observed pixel location  $[u', v']$ . Rearranging (4)

$$\begin{aligned} f_1(u, v) &= (ud_r(u, v) + d_{ud}(u, v)) - u' \\ f_2(u, v) &= (vd_r(u, v) + d_{vd}(u, v)) - v' \end{aligned} \quad (8)$$

and inserting into the general form of the algorithm[68] gives

$$\begin{bmatrix} \frac{\partial f_1(u, v)}{\partial u} & \frac{\partial f_1(u, v)}{\partial v} \\ \frac{\partial f_2(u, v)}{\partial u} & \frac{\partial f_2(u, v)}{\partial v} \end{bmatrix} \begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix} = \begin{Bmatrix} -f_1(u, v) \\ -f_2(u, v) \end{Bmatrix} \quad (9)$$

This method iterates following the algorithm[68]:

- a) At iteration  $i = 1$ , use an initial guess assuming no distortion, i.e.  $[u(i), v(i)]^T = [u', v']^T$
- b) Given  $[u(i), v(i)]$ , solve (9) for  $[\Delta u(i), \Delta v(i)]^T$
- c) Update  $[u(i+1), v(i+1)]^T = [u(i), v(i)]^T + [\Delta u(i), \Delta v(i)]^T$
- d) Evaluate  $|f_j(u(i+1), v(i+1))| \leq \varepsilon, j = 1, 2$  and check for convergence. Where  $\varepsilon$  is the convergence tolerance.
- e) If the tolerance is not met, increase  $i$  and return to step b, otherwise  $[u(i+1), v(i+1)]$  is undistorted pixel value.

### Camera Calibration

From the above sections, it is apparent that there are large number parameters of a camera that must be determined through calibration. Specifically, there are four intrinsic parameters  $(c_x, c_y, f_x, f_y)$  three translation bias terms from

$$\mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \quad (10),$$

three rotation angle terms defining pose from the direct cosine matrix

$$R = R_z(\theta)R_y(\varphi)R_x(\psi) \quad (11)$$

where

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix} \quad (12),$$

$$R_y(\varphi) = \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad (13),$$

and

$$R_z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (14).$$

Additionally, there are the five distortion parameters  $(k_1, k_2, k_3, p_1, p_2)$  that must also be found adding together for a total of fifteen parameters to be calibrated for a single view. It is important to keep a few things in mind:

**Remark 1:** The six extrinsic parameters  $(b_x, b_y, b_z, \psi, \varphi, \theta)$ , change with relative motion of the camera to the global reference frame. Thus using a body reference frame of a moving object over  $k$  image frames for calibration would create  $6k$  parameters to calculate. However for a given resolution and focus, the four intrinsic parameters and five distortion parameters remain constant. As such, for  $k$  image frames of a moving reference frame there are  $6k+9$  total parameters to find. It is worth noting that none of the vision systems presented in this work make use of moving cameras or moving coordinate systems during actual testing, only in the calibration phase.

**Remark 2:** Each of the  $k$  images used in calibration create  $2nk$  constraints based on  $n$  tracked points.

**Remark 3:** Considering the number of parameters from remark 1 and the number of constraints from remark 2 gives the relation of the number points being track and the number of frames used for calibration as  $(n-3)k \geq 5$  (4.5 is rounded to 5 since only full image frames are used).

This calibration accomplished with a series of views of a chessboard, using the inside corners as the points tracked [69]. Figure 2 shows the chessboard used in calibration for the discussed applications and the inside corners tracked for calibration ( $n = 48$ ). At first glance, one might think that only a single image frame is needed for calibration. However, since the all of these points are planar (ie, the chessboard is flat), only four points truly benefit the calibration while the rest allow for consideration of noise and numerical stability[70]. Considering this limitation,  $n = 4$  and a minimum of five image frames with different views of the chessboard are required.

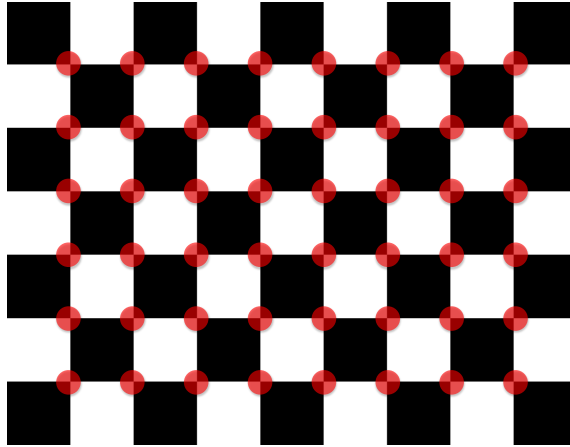


Figure 2: Chessboard used for calibration with inside corners (points tracked) highlighted in red.

Ten image frames seen, ( $k=10$ ), were used to find camera parameters in each application. Since the Intel OpenCV library has a built in function based off Zhang's work[62] for calibration, no new algorithms were developed or written. Figure 4 shows an example of the results of this calibration on image correction, while shows the intrinsic parameters and distortion parameters found through this.



Figure 3: Example of images used in calibration.



Figure 4: Image from camera before (left) and after (right) distortion calibration

Table 1: Intrinsic and distortion camera parameters from calibration

<b>Parameter</b>	<b>Value</b>	<b>Parameter</b>	<b>Value</b>
$f_x$	641.2343	$f_y$	638.6905
$c_x$	337.3445	$c_y$	240.8405
$k_1$	0.1979	$k_2$	-0.9093
$p_1$	0.0035	$p_2$	0.0000

### Stereoscopic Reconstruction

A method for 3D imaging was implemented for use in the shape memory polymer testbed. While it was later determined to be unnecessary for the experiments presented here, the ability to obtain 3D information about an object of interest from a vision system is still a useful skill and will be briefly covered.

The simplest (and ideal) case of stereoscopic reconstruction assumes two perfectly aligned cameras, as shown in Figure 5: image planes are coplanar with parallel principal rays (the ray from the projection center  $O$  passing through the principal point) that intersect at infinity, separated by a known distance  $T$ , and with equal focal lengths  $f$ , and some disparity  $x^l - x^r$  [69].

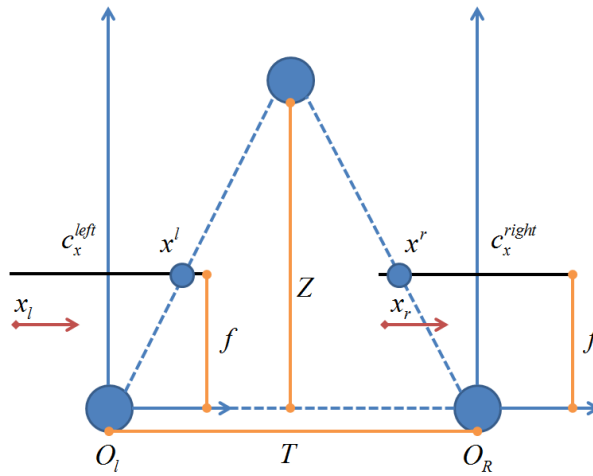


Figure 5: Perfectly aligned stereo setup

From this, the image can be reconstructed [71] as

$$X = \frac{x^l T}{x^l - x^r}, Y = \frac{T(y^l + y^r)}{2(x^l - x^r)}, Z = \frac{fT}{x^l - x^r} \quad (15)$$

Figure 6 shows a general plot comparing disparity to distance, from which it should be apparent that a stereo vision setup's resolution is optimal for objects relatively close to the camera. Increasing the physical distance between the cameras improves the distance at which reliable measurements can be taken, but that is normally limited by physical constraints.

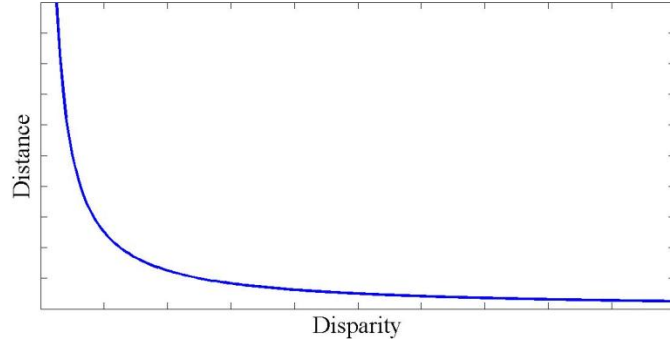


Figure 6: A general plot of disparity versus distance.

This idealized model can be valid for all stereo setups assuming the images from the two cameras are rectified from their current image plane to planes matching the idealized model. There are several methods to rectify the images of two cameras proposed in [69, 72-74] for various setups and applications. [74] presents a clear, general method of rectification which will be summarized as follows:

Given the prospective projective matrixes (PPM) for the left and right camera based on

(1)

$$\tilde{P}_{oi} = [P_i | \tilde{p}_i] = A[R | b], i = l, r \quad (16)$$

new PPMs

$$\tilde{P}_{nl} = \begin{bmatrix} \mathbf{a}_1^T & | & a_{14} \\ \mathbf{a}_2^T & | & a_{24} \\ \mathbf{a}_3^T & | & a_{34} \end{bmatrix} \quad (17)$$

and

$$\tilde{P}_{nr} = \begin{bmatrix} \mathbf{b}_1^T & | & b_{14} \\ \mathbf{b}_2^T & | & b_{24} \\ \mathbf{b}_3^T & | & b_{34} \end{bmatrix} \quad (18)$$

must be found. These new PPMs are constrained by the requirements of the rectified images matching the idealized model including having the same focal plane, creating the constraint

$$\mathbf{a}_3 = \mathbf{b}_3, a_{34} = b_{34} \quad (19)$$

Also, the optical centers, defined as

$$c_i = -P_{oi}^{-1} \tilde{\mathbf{p}}_{oi}, i = l, r \quad (20)$$

of the rectified projections must not change with the rectification

$$\tilde{P}_{ni} \begin{pmatrix} c_i \\ 1 \end{pmatrix} = 0, i = l, r \quad (21)$$

creating the constraints

$$\begin{cases} \mathbf{a}_1^T \mathbf{c}_l + a_{14} = 0 \\ \mathbf{a}_2^T \mathbf{c}_l + a_{24} = 0 \\ \mathbf{a}_2^T \mathbf{c}_l + a_{34} = 0 \\ \mathbf{b}_1^T \mathbf{c}_r + b_{14} = 0 \\ \mathbf{b}_2^T \mathbf{c}_r + b_{24} = 0 \\ \mathbf{b}_3^T \mathbf{c}_r + b_{34} = 0 \end{cases} \quad (22)$$

To be able to use(15), projected vertical coordinates must the same in both frames thus

$$\frac{\mathbf{a}_2^T [X, Y, Z, 1]^T + a_{24}}{\mathbf{a}_3^T [X, Y, Z, 1]^T + a_{34}} = \frac{\mathbf{b}_2^T [X, Y, Z, 1]^T + b_{24}}{\mathbf{b}_3^T [X, Y, Z, 1]^T + b_{34}} \quad (23)$$

which reduces to

$$\mathbf{a}_2 = \mathbf{b}_2, a_{24} = b_{24} \quad (24)$$

with (19). The focal lengths must be parallel, leading to

$$\mathbf{a}_3^T (\tilde{f}_l \wedge \tilde{f}_r) = 0 \quad (25)$$

where  $\tilde{f}_l, \tilde{f}_r$  are the third rows of  $P_{oi}, i = l, r$ . Additionally



$$\mathbf{a}_1^T \mathbf{a}_2 = 0, \mathbf{b}_1^T \mathbf{b}_2 = 0 \quad (26)$$

and setting principal points to  $(0,0)$  gives

$$\begin{cases} \mathbf{a}_1^T \mathbf{a}_3 = 0 \\ \mathbf{a}_2^T \mathbf{a}_3 = 0 \\ \mathbf{b}_1^T \mathbf{b}_3 = 0 \\ \mathbf{b}_2^T \mathbf{b}_3 = 0 \end{cases} \quad (27)$$

New focal lengths  $\tilde{f}_u, \tilde{f}_v$  are arbitrarily defined leading to the constraints

$$\begin{cases} \|\mathbf{a}_1\|^2 \|\mathbf{a}_3\|^2 = \tilde{f}_u^2 \\ \|\mathbf{a}_2\|^2 \|\mathbf{a}_3\|^2 = \tilde{f}_v^2 \\ \|\mathbf{b}_1\|^2 \|\mathbf{b}_3\|^2 = \tilde{f}_u^2 \\ \|\mathbf{b}_2\|^2 \|\mathbf{b}_3\|^2 = \tilde{f}_v^2 \end{cases} \quad (28)$$

Finally the PPMs are constrained by the scaling of

$$\|\mathbf{a}_3\| = 1, \|\mathbf{b}_3\| = 1 \quad (29)$$

These constraints can be arranged into four linear homogenous systems and solved sequentially though singular value decomposition to find the components for desired rectifying PPMs. Once obtained, image frames from each camera can be rectified by

$$\begin{bmatrix} u_{ni} \\ v_{ni} \\ 1 \end{bmatrix} = \tilde{\mathbf{P}}_{ni} \tilde{\mathbf{P}}_{oi}^{-1} \begin{bmatrix} u_{oi} \\ v_{oi} \\ 1 \end{bmatrix}, i = l, r \quad (30)$$

and then mapped using (15).

## CHAPTER THREE: VISION BASED CONTROL SYSTEM APPLICATION – SHAPE MEMORY POLYMER TESTBED

### Point Tracking/Angle Characterization

Initially, a number of colored dots were painted or placed, depending on the size of the sample, along the side of the sample with more being placed along areas of a high curvature as shown in Figure 7. Prior to applying a voltage, the entire images from both cameras were compared to the specific RGB and range values thresholds for each camera and each dot. If all dots are found in user accepted locations, the program stored the points and used a reduced search area centered on the last known location of each dot for the current search (Figure 8). In the event that a dot is not found in its reduced search area, the program searches the entire frame for the dot which typically adds roughly 112ms to runtime for the current system. To reduce background noise, the test bed was covered in white paper before the placement of the sample. Example results of this method can be seen in Figure 9.

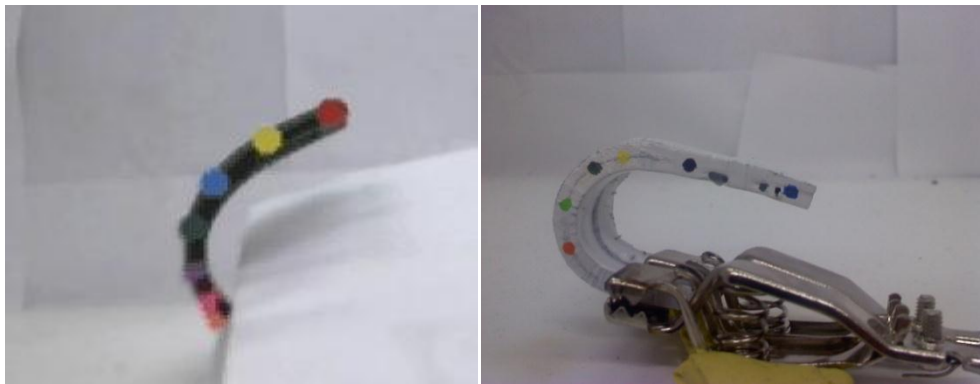


Figure 7: Sample with placed colored dots (left) and painted dots (right)

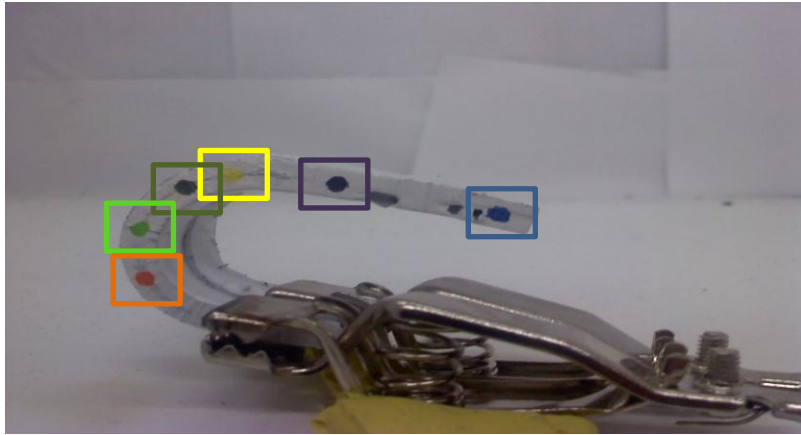


Figure 8: Reduced search area and associated dots

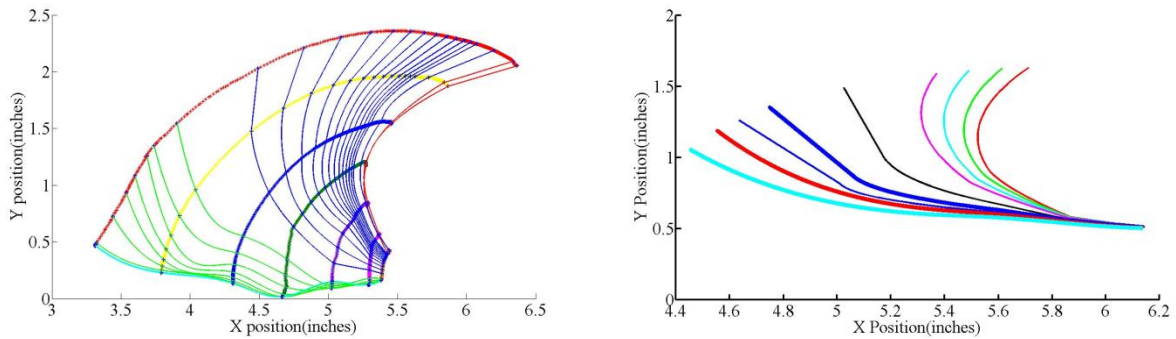


Figure 9: Two different examples of deflection tests tracking multiple dots. (*left*) Dot locations and b-spline curve fitting. (*right*) Linear curve fitting between dots.

While this did give a very clear idea of how the entire sample was reacting, it did have its disadvantages. Tuning thresholds for each sample could be tedious and led to a number of testing failures in which a single dot's RGB values would move out of the threshold region. Additionally, this many data points along samples was unnecessary in the proposed control scheme.

To simplify this, the angle of tip deflection (deflection angle),  $\omega$ , was measured. This was accomplished by using the topmost two dots and finding the angle between the vector they create and the horizon, or

$$\omega = \tan^{-1} \left( \frac{y_1 - y_2}{x_1 - x_2} \right) \quad (31)$$

Figure 10 shows an example of this.

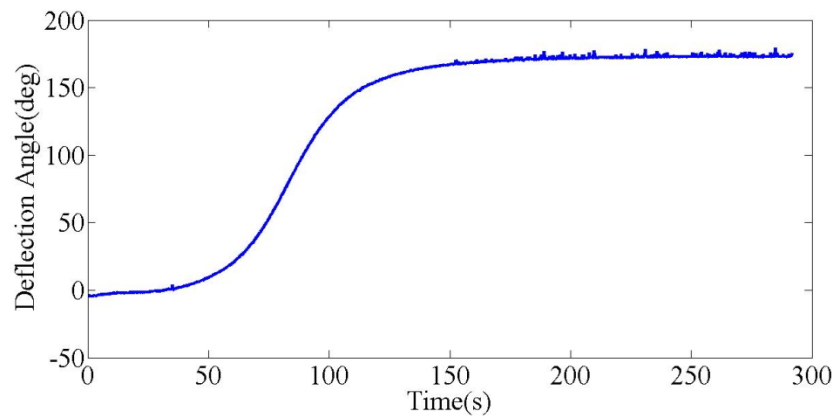


Figure 10: Example of sample test measure tip deflection angle (deflection angle)

### Hardware Setup

The testbed, seen in Figure 11, designed for the testing and analysis of memory polymers consisted of two Logitech webcams fixed to a static table, a voltage regulator, a laptop, and an Arduino Uno. A general flow diagram is provided in Figure 12 and will be described below.

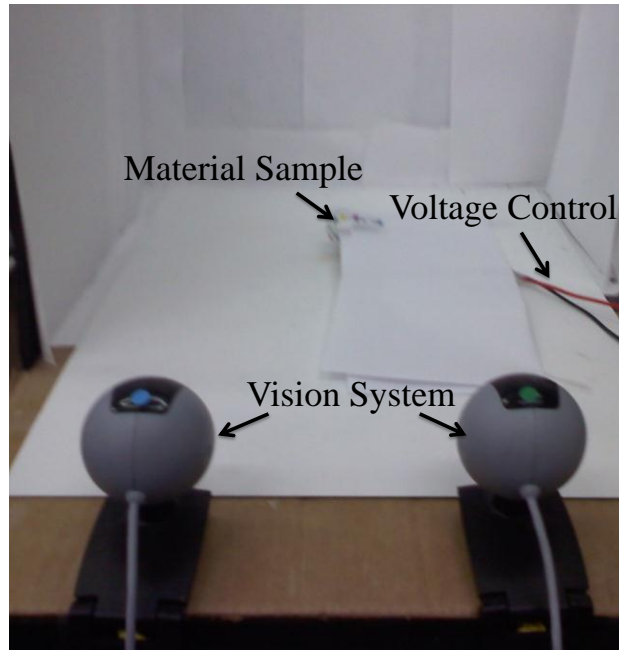


Figure 11: Testbed for memory polymers

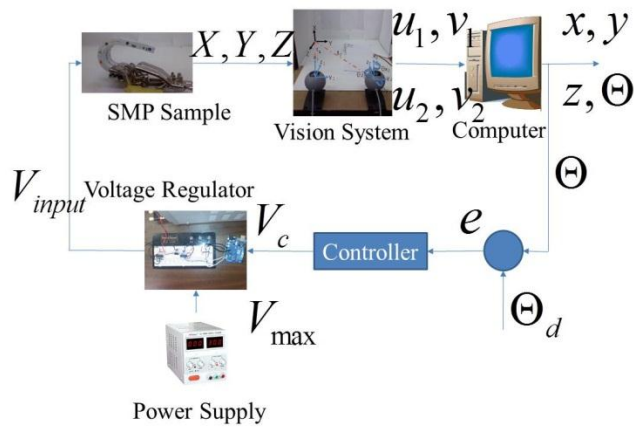


Figure 12: Data flow of the SMP testbed

Image data was obtained through the camera system and processed by the laptop which in turn would determine the input voltage. This desired voltage, as determined by the control

algorithm, is sent to the Uno over a standard serial connection and generated through the voltage regulator

### PID Control

Based on the error signal  $e(k)$ , the following PID controller is designed and three gains  $k_p, k_i$ , and  $k_d$ , a PID controller applies and input such that

$$u(k) = P(k) + I(k) + D(k) \quad (32)$$

or

$$u(k) = k_p e(k) + k_i \int_0^k e(\tau) d\tau + k_d \dot{e}(k) \quad (33)$$

For the discrete system with a time-step  $T_s$ , it is useful to define the integral portion using the trapezoidal rule[75]

$$I(k) = k_i \left( \int_{k-1}^k e(k) + \int_0^{k-1} e(k) \right) = k_i \left( T_s \frac{e(k) + e(k+1)}{2} \right) + I(k-1) \quad (34)$$

and the derivative portion using backward difference as

$$D(k) = k_d \frac{e(k) - e(k-1)}{T_s} \quad (35)$$

The proportional value,  $P(k)$ , remains unchanged.

In many systems, such a controller would achieve the desired response with the appropriate gains. Since the response of the material is one-directional and cannot recover from overshoot, a precaution was taken with the integral controller. In similar situations, it is common to have some kind of anti-windup scheme [76-78]. To prevent windup that could not be

compensated by the other gains, a reset function that would set the iterative sum back to zero based on the following

$$I(k) = P_I \left[ k_i T_s \frac{1}{2} [e(k) + e(k-1)] + I(k-1) \right] \quad (36)$$

$$P_I = \begin{cases} 1 & \text{if } k_i T_s \frac{1}{2} [e(k) + e(k-1)] + I(k-1) < V_{i\max} \\ 0 & \text{if } k_i T_s \frac{1}{2} [e(k) + e(k-1)] + I(k-1) \geq V_{i\max} \end{cases} \quad (37)$$

is designed. Ideally  $V_{i\max}$  would be chosen such that when  $e(k) \sim 0$ , the integral gain alone cannot cause a voltage that would overcome the heat loss due to convection given by[79]

$$Q = c_p m \frac{dT}{dt} = h(T_\infty - T_m) + \frac{V_{i\max}^2}{R} \quad (38)$$

where  $Q$  is the overall heat transfer,  $c_p$  is the specific heat of the sample,  $m$  is the mass of the sample,  $h$  is the coefficient of convection between the exposed surface of the material with the surface temperature of  $T_m$  and the surrounding air of temperature  $T_\infty$ . Giving the constraint

$$V_{i\max} \leq \sqrt{Rh|T_\infty - T_m|} \quad (39)$$

and iteratively updated based on this. However, it was found that a constant value proved adequate.

Figure 13 shows the final results of a PID controller after iterative tuning of its parameters ( $k_p = 20, k_i = 15, k_d = 4$ ). Each sample was first heated with a voltage of 25v until malleable, deformed under loading to  $\sim 180$  degrees from the memorized state, and allowed to cool back to room temperature before testing. These had a reasonably fast response of  $\sim 90$

seconds and could be repeatedly commanded to the desired angle within 5%. This constant, precise repeatability is important for any kind of future actuator or mechanism application that this material could be applied to.

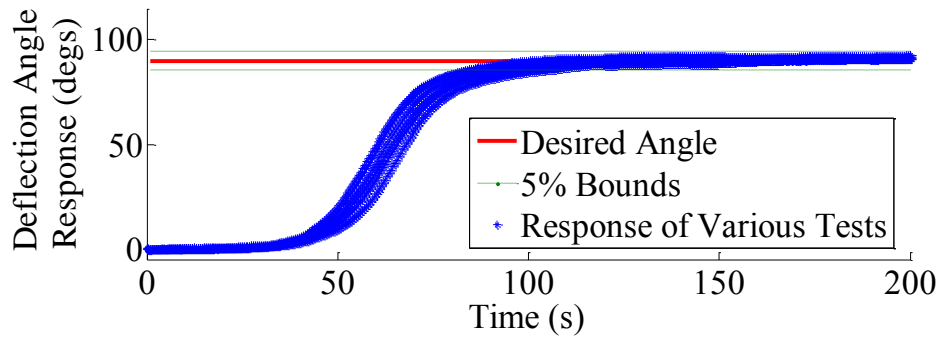


Figure 13: Control response for ten different samples

In a typical response, the proportional gain drove the response of the sample in each test for around 50 seconds (Figure 14). Saturation control and the resetting function in the iterative gain kept this initial input acting as the standard step input seen in most SMP testing.



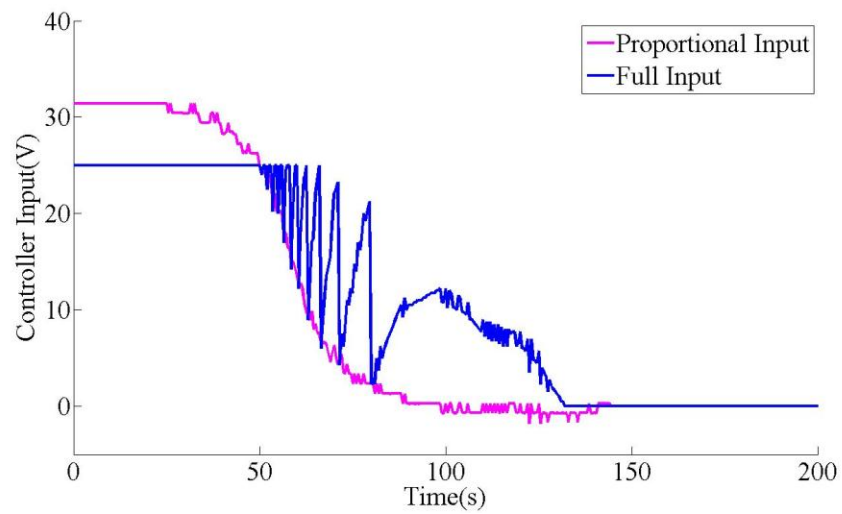


Figure 14: Typical proportional gain input

With time, however the error signal becomes small enough to almost negate the proportional gain and the integral gain becomes the dominant driver of the voltage (Figure 15).

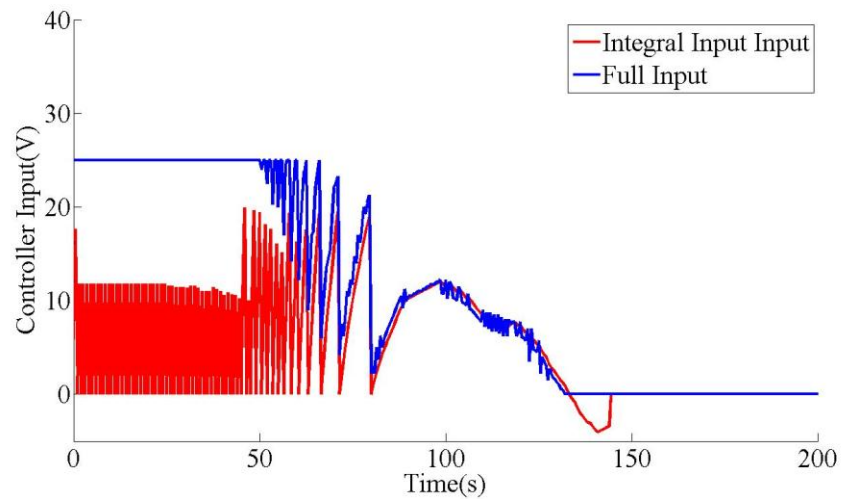


Figure 15: Typical integral gain input

It was found that in the pursuit of a fast response, a small derivative gain was desirable thus making its contribution to the voltage input minimal (Figure 16). Larger values increased the response time by up to 40 seconds, while only decreasing the overshoot 2 degrees (best case). Figure 17 displays a typical power requirement of the controller for a material with a  $100\Omega$  resistance.

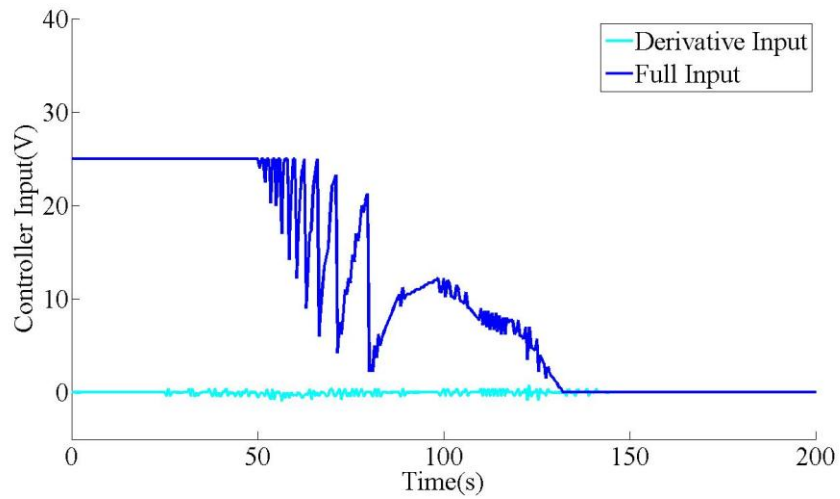


Figure 16: Typical derivative gain input

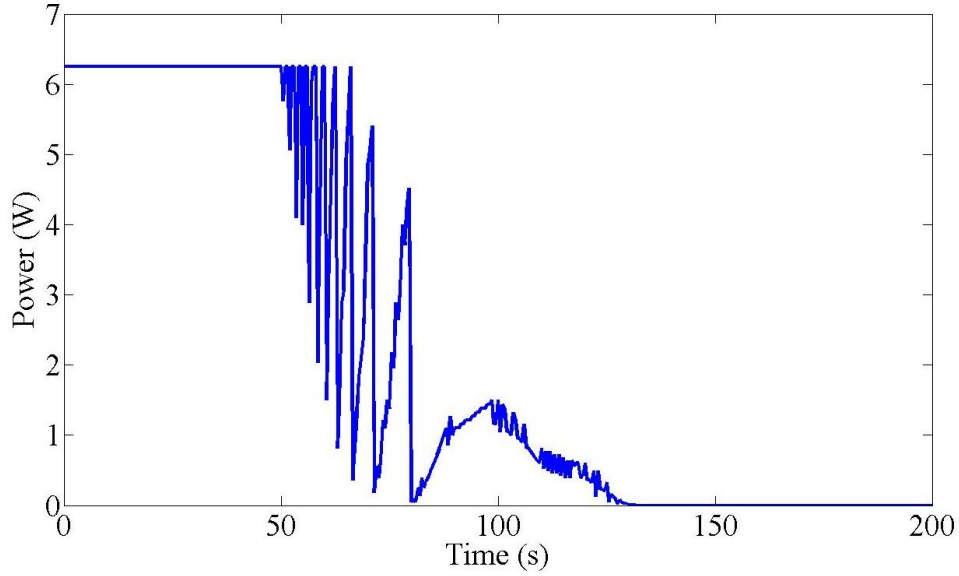


Figure 17: Typical power requirement of the PID controller

### System Modeling

Being able to model the system being controlled is very beneficial to the development of control for the system. It provides the basis for stability analysis as well as a number of control schemes[80, 81]. While there are a number of excellent works that model shape memory polymers at the micro [82-84] and macro[85, 86] levels, they require case by case validation or a time intensive finite element analysis. Additionally, a detailed constitutive model is not needed for precise deflection control, as will be shown below.

A general linear model is proposed: The dynamic response of the sample SMP under a single voltage input is described by the following discrete equation

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, u_k), k = 0, \dots, N, \dots \quad (40)$$

and the measurement is modeled given by

$$\mathbf{y}_k = h(\mathbf{x}_k), k = 0, \dots, N, \dots \quad (41)$$

where the state vector  $\mathbf{x}_k = [x_{i,k}, y_{i,k}, z_{i,k}, T_{i,m,k}, T_\infty]^T$ ,  $i = 1, 2, \dots, n_v$  of the system are positions of each of the  $n_v$  interested dots on the surface of the SMP, the temperature of the sample at each point  $T_{m,i,k}$ , and the temperature of the surrounding room  $T_\infty$ . The measured output is a nonlinear function equating the position of points, which are along the side of the material as described in, to an angle  $\Theta$ . Also, the input  $u(k)$  is the square of the voltage applied on one end of the sample SMP through the regulator. These states arise from the heat transfer (38) and an effort to find a linear relation sample temperature and sample motion.

The SMP material is nearly homogenous. Based on a heat transfer analysis of the material assuming (1) heat loss through the test-bed and radiation is negligible and (2) uniform heating ( $T_{m,i,k} \rightarrow T_{m,k}$ ) of the sample through electrical resistive heating, the response of the SMP shape changes can be approximated by

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k \quad (42)$$

so for a sample with a single node

$$\begin{bmatrix} x_{1,k+1} \\ y_{1,k+1} \\ z_{1,k+1} \\ T_{m,k+1} \\ T_{\infty,k+1} \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & P(T_{m,k})A_{1,4} & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & P(T_{m,k})A_{2,4} & 0 \\ A_{3,1} & A_{3,2} & A_{3,3} & P(T_{m,k})A_{3,4} & 0 \\ 0 & 0 & 0 & 1 + \frac{h}{cm}T_s & \left(-\frac{h}{cm}\right)T_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ y_{1,k} \\ z_{1,k} \\ T_{m,k} \\ T_{\infty,k} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{T_s}{cmR} \\ 0 \end{bmatrix} V^2 \quad (43)$$

and for a more general case

$$A = \begin{bmatrix} A_{1,1} & \cdots & A_{1,3n_v} & P(T_{m,k})A_{1,3n_v+1} & 0 \\ \vdots & \ddots & \vdots & \vdots & 0 \\ A_{3n_v,1} & \cdots & A_{3N,3n_v} & P(T_{m,k})A_{3n_v,3n_v+1} & 0 \\ 0 & 0 & 0 & 1 + \frac{h}{cm}T_s & -\left(\frac{h}{cm}\right)T_s \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (44)$$

and

$$B = [0, \dots, 0, T_s / cmR, 0]^T \quad (45)$$

where  $A$  is the exposed area of the material of mass  $m$ ,  $h$ ,  $c$ , and  $R$  are the coefficient of convection, specific heat and resistance of the sample, respectively.  $A$  and  $B$  are constant matrices related to state and input respectively while  $A_{ij}; (i=1:n_v, j=1:n_v+1)$  are constants to be determined through experiments and system identification.  $P_T(T_{m,k})$  is a Piecewise function to assist in simulating the dependency of the SMP's angle response on internal temperature being at least equal to the glass transition temperature

$$P_T(T_m) = \begin{cases} 0 & \text{if } T_m < T_g \\ 1 & \text{if } T_m \geq T_g \end{cases} \quad (46)$$

Obviously, this system is severely under-actuated considering the single input. In an effort to reduce this under-actuation from a controllability perspective, the system is further reduced to

$$\begin{bmatrix} \theta_{k+1} \\ T_{m,k+1} \\ T_{\infty,k+1} \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2}P(T_{m,k}) & 0 \\ 0 & 1 + \frac{sh}{cm}T_s & \left(-\frac{sh}{cm}\right)T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_k \\ T_{m,k} \\ T_{\infty,k} \end{bmatrix} + \begin{bmatrix} 0 \\ T_s \\ 0 \end{bmatrix} \frac{1}{cmR} V^2$$

### Identification

For more advanced methods of control, it becomes necessary to obtain a representation of the system through testing. A blackbox model was generated using the input (voltage) and output (deflection angle) data from a series of step response tests and the system identification toolbox in matlab using prediction error minimization (PEM). PEM, as the name implies, generates a linear state space model

$$\begin{aligned}\hat{\mathbf{x}}(k+1) &= F\hat{\mathbf{x}}(k) + Gu(k) \\ \hat{y}(k) &= H\hat{\mathbf{x}}(k)\end{aligned}\tag{47}$$

that minimizes the different between the predicted model  $\hat{y}(k)$  and the measured output  $y(k)$  for the measured input  $u(k)$ . This method alone cannot model the memory effect seen in samples, so only the system response after  $T_m \geq T_g$  was modeled. This was accomplished by adjusting the input  $\hat{u}(k)$  used in the modeling algorithm

$$\hat{u}(k) = \begin{cases} 0 & \text{if } \dot{\theta} = 0 \\ u(k) & \text{if } \dot{\theta} > 0 \end{cases}\tag{48}$$

As can be seen in Figure 18, the model generated using this data had a similar response for the higher voltage range. Since this is a blackbox model, it should be noted that the state vector within the model are not necessarily related to the states described in the ideal model suggested in System Modeling and may not be measureable. However, the states of a sample as relating to the model can be estimated by the model itself for a given input.

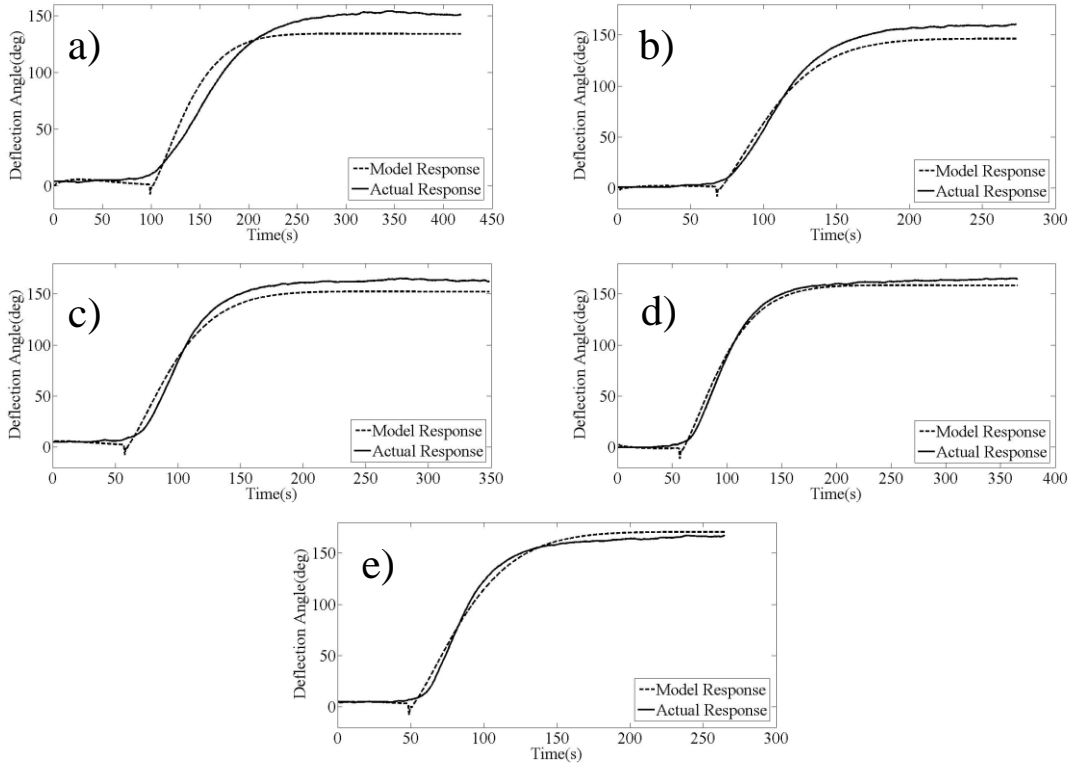


Figure 18: Responses of models compared to measured results for a) 22v input b) 24v input c) 25v input d) 26v input e) 28v input.

### LQT Control

Using the model found above, a linear quadratic tracking(LQT) controller[80, 81] was designed. This creates a state feedback input

$$u = -K\mathbf{x} \quad (49)$$

that minimizes the cost function

$$J = \int_0^{\infty} (\mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}) dt \quad (50)$$

where  $Q$  and  $R$  are tunable and the gain  $K$  is based off of

$$K = R^{-1} (B^T S + N^T) \quad (51)$$

and  $S$  is obtained by solving the Riccati equation

$$A^T S + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = 0 \quad (52)$$

State tracking is accomplished with the addition of

$$u = -K\mathbf{x} + R^{-1}B(SBR^{-1}B^T - A)C'Q\mathbf{x}_d \quad (53)$$

where  $\mathbf{x}_d$  is the desired final state vector for the desired output.

The LQT controller was applied to the model created above with a desired deflection angle of 90 degrees. Figure 19 shows the simulation results of the model response to this controller and Figure 20 displays the voltage inputs obtained from the controller. It is worth noting that while the high voltage may seem out of place, the model has reached a steady-state value satisfied by

$$(F - I)x(k) + Gu(k) \sim 0 \quad (54)$$

It is worth noting that with a real sample, the voltage would be turned off after the desired steady deflection is obtained.



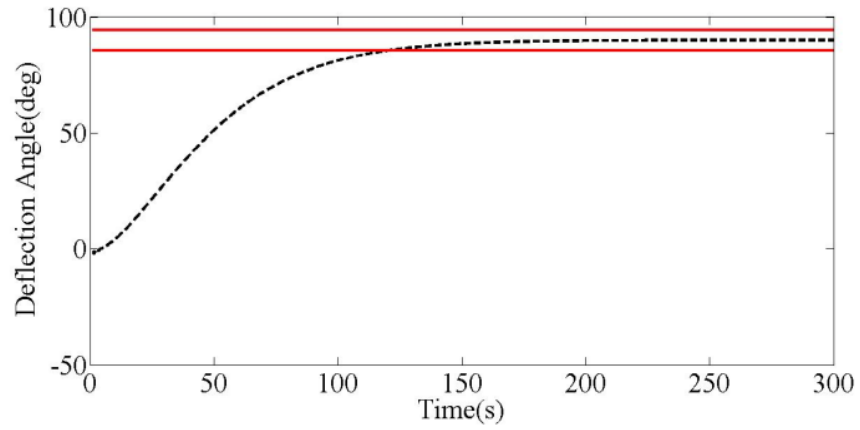


Figure 19: Deflection response of model to LQT controller (*black*) and five percent error bounds (*red*)

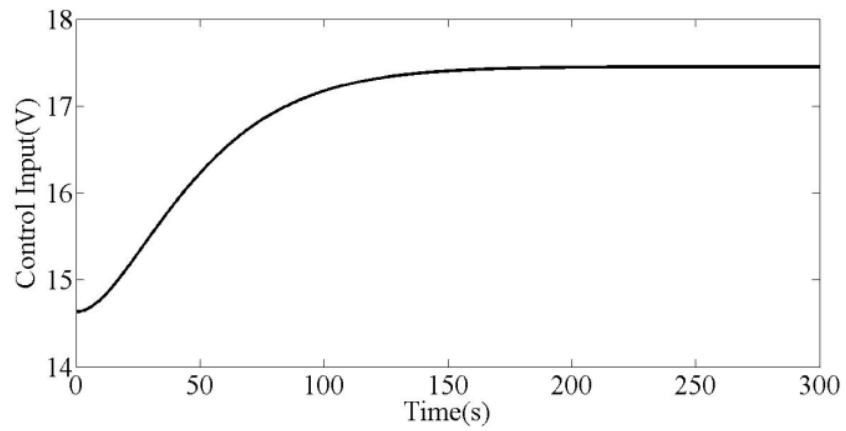


Figure 20: Voltage input to model from LQT controller

From these simulation results, the LQT control method may be a viable solution for precise deflection error control.

### User-Interface

For the majority of the testing with this testbed, a command-prompt interface programmed in C was used. It offered semi-autonomous calibration, stereoscopic reconstruction, and met all data acquisition and control needs. For a new user, however, it was non-intuitive and, from a processing standpoint, had a number of unneeded features. With that in mind, a new interface was created in labview.

Seen in Figure 21, this interface provides a real time view of the sample being measure, real time plots of current readings, material resistance calculations, and material temperature readings. It also has overcurrent protection and permits the user to actively update RGB thresholds during testing. Stereoscopic reconstruction, and thus the second camera, was not carried over to this new UI as the capability was deemed not necessary.

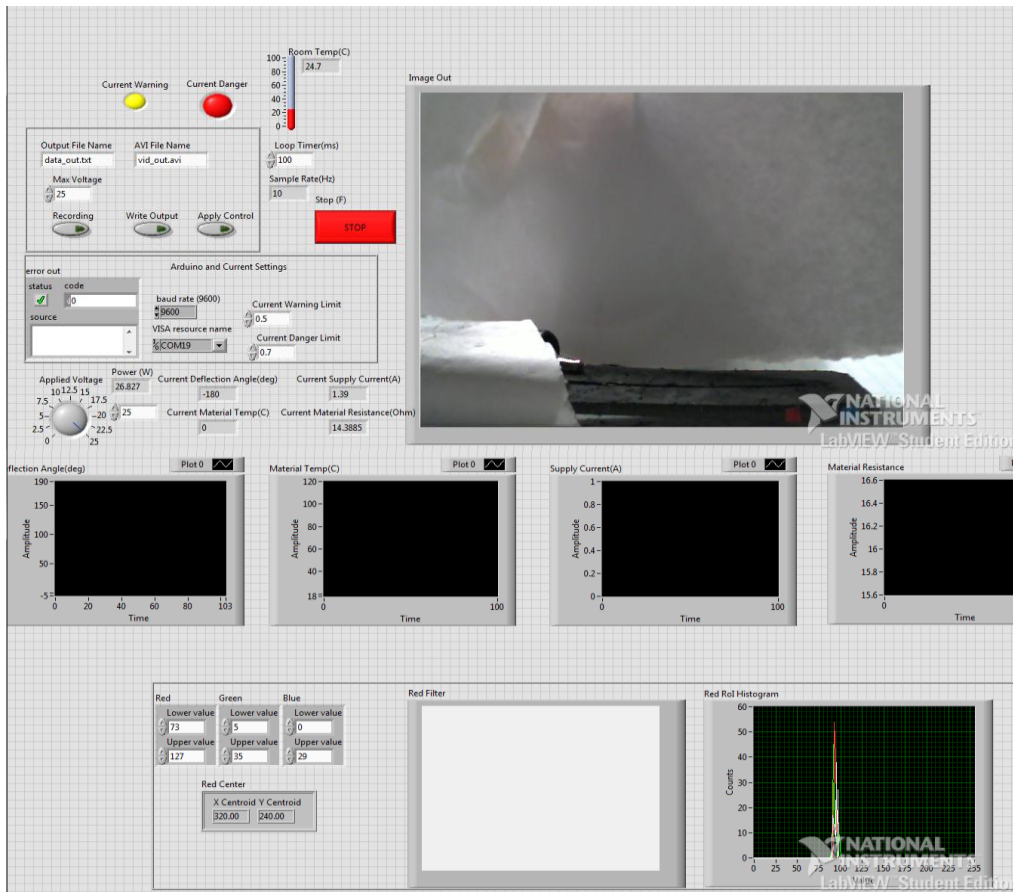


Figure 21: User interface for SMP testbed

## CHAPTER FOUR: VISION BASED CONTROL SYSTEM APPLICATION- MOBILE ROBOT TESTBED

### BVMC

A recently developed B-spline augmented virtual motion camouflage (BVMC) is applied to solve the nonlinear constrained trajectory planning problem[50]. For our test example, the performance index is

$$J = \int_{t_0}^{t_f} dt \quad (55)$$

and the nonlinear dynamics is shown in Eq. (67). This optimization is constrained by the physical dimensions of the tested  $0 \leq x \leq 640$  pixels and  $0 \leq y \leq 480$  pixels, and the hardware limitations of the motors on the robot are  $0 \leq V \leq V_{\max}$  and  $|\omega| \leq \omega_{\max}$

Obstacle avoidance is considered with the additional inequality constraints

$$\|\mathbf{r}_a - \mathbf{r}_{obs,i}\| > R_{obs,i} + R_a, i = 1, \dots, n_{obs} \quad (56)$$

given a robot position  $\mathbf{r}_a = [x, y]^T$ , obstacles' center positions  $\mathbf{r}_{obs,i} = [x_{obs}, y_{obs}]^T, i = 1, \dots, n_{obs}$ , the radius of obstacle  $R_{obs,i}$ , and a buffer to account for the form factor of the robot  $R_a$ .

The basic concept of the BVMC method is that the vehicle's trajectory is optimized in a varying manifold constructed by a reference point  $\mathbf{r}_{ref} = [x_{ref}, y_{ref}]^T$  and a virtual prey motion  $\mathbf{r}_p = [x_p, y_p]^T$  according to

$$\mathbf{r}_a = \mathbf{r}_{ref} + v(\mathbf{r}_p - \mathbf{r}_{ref}) \quad (57)$$

The one dimensional vector  $v$ , known as the path control parameter (PCP), is optimized within the manifold. The benefit of this method is that a rapid convergence to an optimal solution

can be achieved in the presence of severe constraints and large numbers of obstacles. To achieve the optimal solution, the virtual prey and the reference point, i.e. the varying manifold, have to be optimized. In the BVMC approach, the virtual prey is represented using the B-spline curve, of which the control points will be optimized. As mentioned in Wavefront Path Planning, to increase the convergence speed, the result from the wavefront path planning algorithm is used as the initial guess of the virtual prey motion.

#### Hardware in the Testbed

The laptop computer hosts the software for image processing, path optimization, wireless communication, and issuing commands to the robots. Everything was programmed in the Matlab 2010b environment. Image acquisition and processing are assisted by the image acquisition toolbox and image processing toolbox, respectively, while the optimization toolbox is used in trajectory optimization. Also the laptop is connected with the webcam to capture images.

For the testbed design for UCF, a Logitech webcam was mounted over the test area in a unique housing. The test area was covered in white paper to reduce image noise and obstacles were cut out of foam.

At the Air Force Research Lab, Logitech webcams were mounted to booms and positioned over the test area. The testbed floor was designed in a modular, puzzle-piece fashion as seen in Figure 22 and cut from wood. Obstacles were also cut from wood.

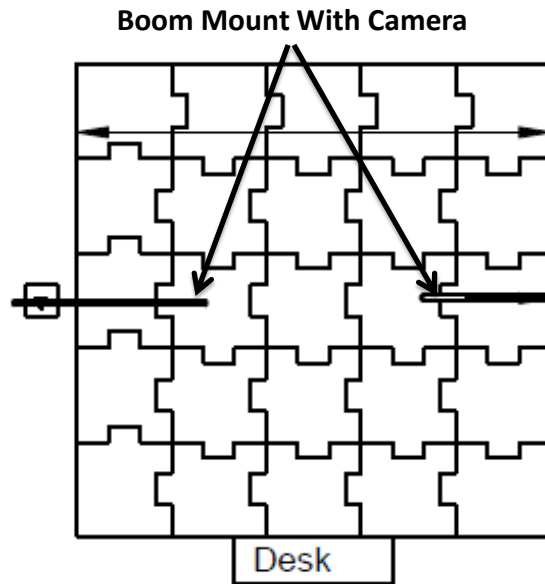


Figure 22: Testbed design for AFRL mobile testbed

### Robot Platform

In the pursuit of being able to support the testing of new algorithms designed for multi-agent systems, a dynamic array of LEGO<sup>®</sup> robot platforms are used. The physical structure of the robot can be easily rebuilt to meet different physical and mission requirements, and the overall cost of the system is considerably low. It is worth noting that regardless of the physical form applied to the robots, the processor and software structure of the system remains the same, which reduces the complexity of changing between experiments.

The robot onboard processor supports four sensor input ports (though this can be expanded with a multiplexer) and three output motor ports. A broad range of sensor packages including gyroscopes, accelerometers, sonar, and IR can be integrated into the robot platform.

---

<sup>®</sup> LEGO is a trademark and/or copyright of the LEGO Group

Each robot is able to send and receive data to and from its associated laptop over an onboard Bluetooth<sup>®</sup> connection.

As can be seen in Figure 23, the robot design incorporates a dual motor, treaded drive system with a tray structure placed over the main body of the robot. This tray provides both a level surface to affix image templates used in the vision system, to be explained later, and room to expand the number of sensors available in later experiments. This design had a measured maximum translational speed of  $V_{\max} = 24.4\text{cm}/s$  and a maximum rotational speed of  $\omega_{\max} = 1.5\text{rad}/s$  for the 100% differential drive power settings. Different color dots on top of the robots (Figure 24) are used by the vision processing algorithm to differentiate the robots and calculate the heading information.

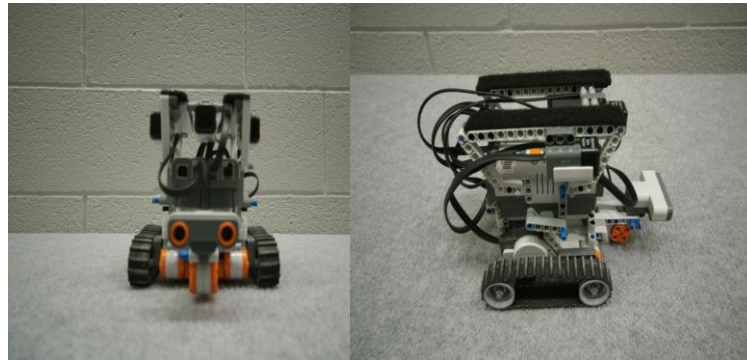


Figure 23: Robot used in testing algorithms

---

<sup>®</sup> Bluetooth word mark and logos are registered trademarks owned by Bluetooth SIG, Inc.

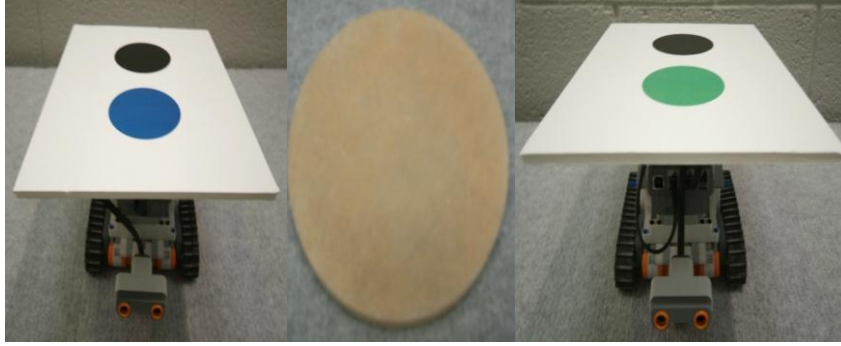


Figure 24: Vision system identifiers used by robots (left and right) and example obstacle (middle).

### Robot Localization and Obstacle Detection

Robot localization and obstacle detection are accomplished through almost identical means using an overhead camera, capturing 480x640 resolution images at rates up to 30fps. In either case, a normalized cross correlation matrix (NCC)  $\gamma \in R^{480 \times 640}$  is created using a grayscale image from the vision system,  $H_{gs} \in R^{480 \times 640}$ , and a grayscale template of the object of interest,  $f \in R^{M \times N}$ , such that

$$\gamma(u, v) = \frac{\sum_{x,y} [H_{gs}(x, y) - \bar{H}_{u,v}] [f(x-u, y-v) - \bar{f}]}{\left\{ \sum_{x,y} [H_{gs}(x, y) - \bar{H}_{u,v}]^2 \sum_{x,y} [f(x-y, y-v) - \bar{f}]^2 \right\}^{0.5}} \quad (58)$$

, where  $\bar{f}$  is the mean of the grayscale template and  $\bar{H}_{u,v}$  is the mean of the grayscale image under the template shifted to  $[u, v]$ .

Object locations are determined by the local maximums of  $\gamma$  passing a set tolerance. The results are illustrated in Figure 25 and Figure 26 for the obstacle detection and the robot



localization, respectively. It is worth noting that because all obstacles currently used in the testbed are homogenous, this is all that is required to obtain the location and radius of the obstacles  $[x_{obst}, y_{obst}, r_{obst}]_i, i = 1, \dots, n_{obst}$ , where  $n_{obst}$  is the number of obstacles.

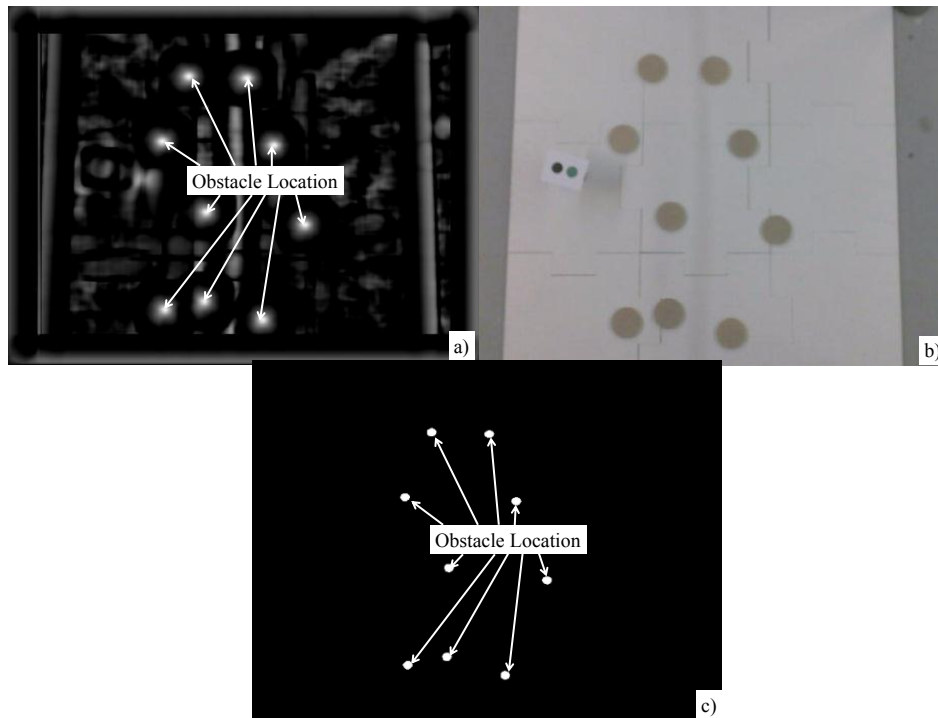


Figure 25: Visual output of a NCC using the obstacle template. a) After pass of NCC b) Original image c) after thresholding

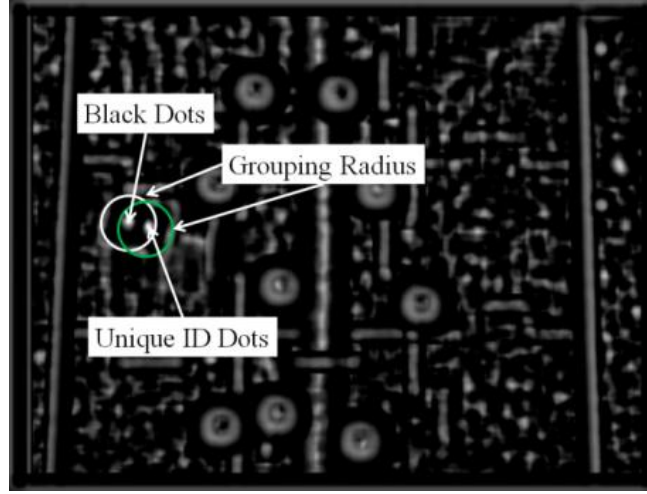


Figure 26: Visual output of a NCC using the robot dot template before thresholding. Based off the same original image seen in Figure 25.a.

In order to fully determine the  $n_r$  robots' locations and headings,  $[x_{rob}, y_{rob}, \theta_{rob}]_i, i = 1, \dots, n_r$ , more computation is require. The average RGB values of the areas surrounding the dot locations,  $[x_{dot}, y_{dot}]_i, i = 1, \dots, 2n_r$ , are found and compared to the predefined thresholds to determine the sets of unpaired colored dots  $x_{cd}, i = 1, \dots, n_r$ , and  $x_{bd}, i = 1, \dots, n_r$  as seen in Figure 26. Black dots and colored dots, the latter of which acts as a unique ID for each robot, are then paired by proximity. It is worth noting that as more robots are involved in the system the more difficult it becomes to tune a specific and unique RGB threshold for each robot. In that case a different identification method such as the ones in [87-89] would be necessary.

The position for each robot is then determined as

$$\begin{aligned} x_{rob,i} &= (x_{cd,i} + x_{bd,i}) / 2 \\ y_{rob,i} &= (y_{cd,i} + y_{bd,i}) / 2 \end{aligned}, i = 1, \dots, n_r \quad (59)$$

, while the heading is found using

$$\theta_{rob,i} = \arctan\left(\frac{y_{cd,i} - y_{bd,i}}{x_{cd,i} - x_{bd,i}}\right), i = 1, \dots, n_r \quad (60)$$

When programmed in MATLAB, these algorithms can be used to process images at 100Hz for the robot localization, and 7.4Hz for the obstacle detection, respectively. Therefore all vision data is updated every 0.14 seconds (7.4Hz).

It is worth noting that the method of robot localization and obstacle detection could replace the RGB thresholding method described in Point Tracking/Angle Characterization to improve the sample rate seen in the SMP testbed. Since new tracking points are not added mid experiment in that testbed, the full 100Hz sampling could be realized.

#### Path Tracking

Here a tracking controller is designed to calculate proper motor commands for robots to track the generated optimal path  $(x_i^*, y_i^*), i = 1, \dots, n_r$  via the BVMC method. If the current position is  $(x_{rob,i}, y_{rob,i}), i = 1, \dots, n_r$ , the rotation command is given based on

$$\Delta\theta_{rob,i} = \arctan\left(\frac{y_i^* - y_{rob,i}}{x_i^* - x_{rob,i}}\right), i = 1, \dots, n_r \quad (61)$$

, and then a translational command is computed as

$$\Delta x_{rob,i} = \sqrt{(x_i^* - x_{rob,i})^2 + (y_i^* - y_{rob,i})^2}, i = 1, \dots, n_r \quad (62)$$

Rotational and translational power commands for a time step  $\Delta t$  can be found by

$$g_{R,i} = \frac{\Delta\theta_{rob,i}}{\Delta t C_{R,i}}, i = 1, \dots, n_r \quad (63)$$

and

$$g_{T,i} = \frac{\Delta x_{rob,i}}{\Delta t c_{T,i}}, i = 1, \dots, n_r \quad (64)$$

where

$$c_{R,i} = \frac{r}{d} (c_{l,i} + c_{r,i}), i = 1, \dots, n_r \quad (65)$$

$$c_{T,i} = \frac{r}{2} (c_{l,i} + c_{r,i}), i = 1, \dots, n_r \quad (66)$$

This has proven to provide adequate tracking of the planned trajectories, as can be seen in Figure 27, and typically have a maximum tracking error of around 4 pixels or 1.1 cm.

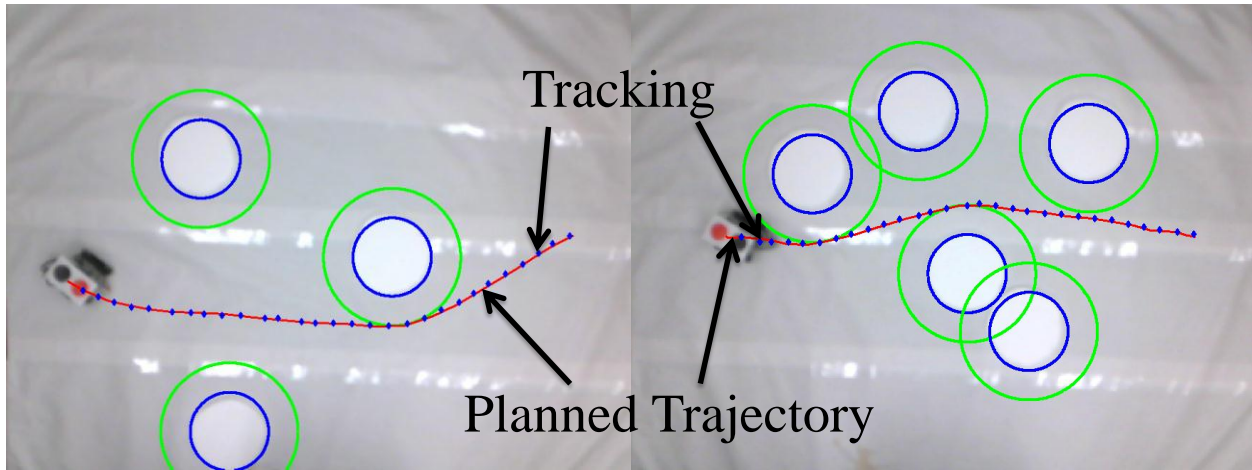


Figure 27: Examples of path tracking (*blue diamonds*) for given trajectories (*red*)

#### Automatic Robot Parameter Calibration

For a differential drive robot, such as the ones used in the testbed, and assuming a case of no slipping, the robot can be controlled by the turn rate of  $\omega$  and the translational velocity of  $V$ , and the motion is governed by the following dynamics [90]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{bmatrix} V + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (67)$$

$V$  and  $\omega$ , depicted in Figure 28, can be generated based on the left and right wheel angular speeds [91]

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2}(\omega_l + \omega_r) \\ \frac{r}{d}(-\omega_l + \omega_r) \end{bmatrix} \quad (68)$$

with the wheel radius  $r$  and the distance between wheels  $d$ , measured in  $cm$ . The power level commands  $[\mathcal{G}_l, \mathcal{G}_r]$ , used in the robot firmware, for generating desired angular velocities of the wheels can be computed by

$$\mathcal{G}_i = \omega_i / c_i, \quad i = l, r \quad (69)$$

where the constants  $c_i$ , can be found through calibration.

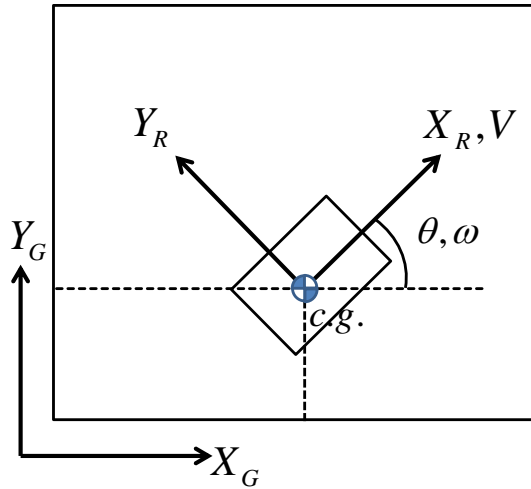


Figure 28: Body and global reference frames used in testbed.

A program was developed for a quick calibration and autonomous calibration. A series of rotation and translation commands are sent to a robot for known time steps and the responses (i.e.  $\Delta x, \Delta \theta$ ) are measured. An example of this can be seen in Figure 29. From these measurements, constants can be found through a least squares approach. Table 2 shows the measured and calculated values from the example. It is worth noting that it from equations (65) and (66), isn't necessary to measure the robot's wheel radius and width as they can be grouped into the constants of rotation and translation.

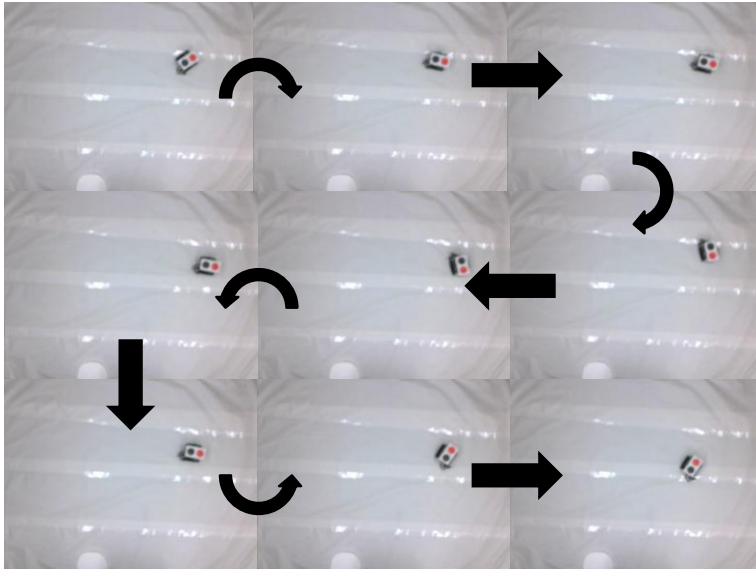


Figure 29: Example of automatic calibration for a robot. Commands alternate between positive and negative rotations and translations.

Table 2: Measured and Calculated Values for the Example Calibration

Parameter	Value
$r$	2.54cm
$d$	15.24cm
$c_r$	0.09396
$c_l$	0.10396

## Graphic User Interface

The GUI created (Figure 30) for this testbed acts as a straight forward interface between the user and the hardware to aid in the fast setup, running, and viewing of different experiments. There are four main functionalities involved in the GUI.



Figure 30: Overall view of the GUI

**Functionality 1: Connection Management** - As seen in Figure 31, connections to both the vision system and the robots can be easily toggled. Multiple robot connections can be managed with a dropdown menu.



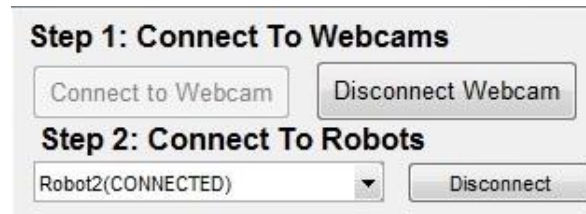


Figure 31: Easy to use connection management of both the vision system and the robot connections

**Functionality 2:** Environment Definition - Figure 32 shows how the user is able to select which obstacles detected through the vision system are “seen” by the robots. This feature also can simulate cases with limited environment knowledge or a changing environment. This is typically done by breaking the run into multiple sections, labeled ‘first’ and ‘second’, etc., in the figure, and assigning obstacles to the section where it is considered. This method of detection then assignment can be overridden by the user so that environmental changes are continually searched for an updated.

Figure 32 shows the overhead vision system and displays the overlays of the detected obstacles, their buffers, and trajectory information to be described in Functionality 3. The buffer overlays are based on half the robots length. Obstacle detection is accomplished through the method to be described in Robot Localization.

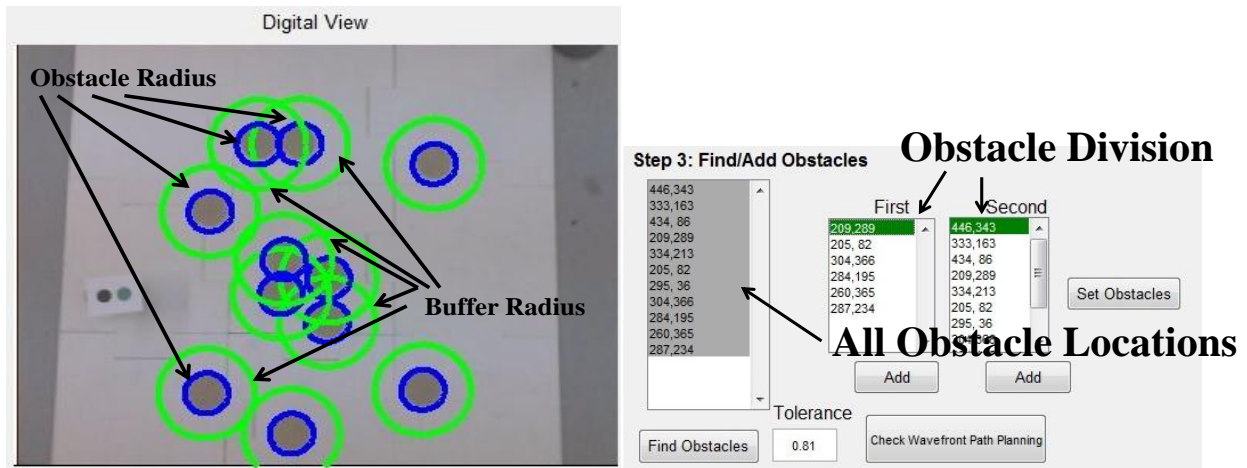


Figure 32: GUI showing obstacles detected and being divided into the two sections of the run

**Functionality 3: Result Display** - Figure 33 shows how the planned trajectories and trajectory tracking results are plotted in the upper right of the GUI. This gives the user instantaneous knowledge of how well the algorithm is performing.

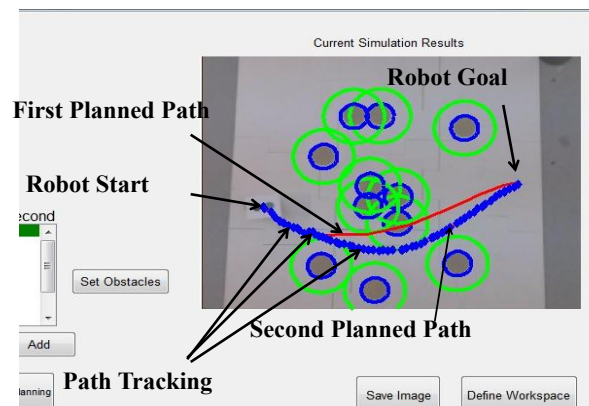


Figure 33: GUI showing path planning and path tracking results of a typical run

**Functionality 4: Robot Viewer** - A second window of the GUI is shown in Figure 34. This “Robot Viewer” enables the user to monitor the battery level and sensor data, as well as,

adjust robot trajectory target locations, calibrate sensors, and select each robot's method of localization via the vision system.

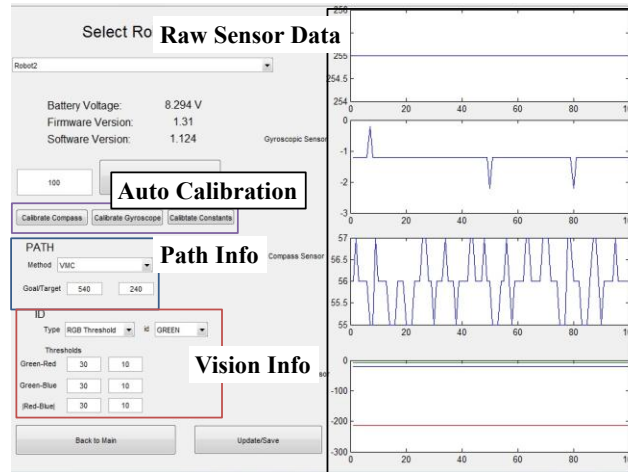


Figure 34: Robot Viewer displaying raw sensor data, vision system information, and path generation info

### Wavefront Path Planning

Many methods have been proposed for finding optimal trajectories [92]. However, many of these require a good initial guess, so that a quick convergence can be achieved, especially when there are many obstacles. Here a method known as the wavefront method is implemented[93]. It does not consider robot constraints, dynamics, or initial headings, and the only goal is to find an obstacle-free trajectory that will accommodate the robot's size.

**Phase 1.** The workspace is decomposed into a number of squares based on the robot's largest dimension. For example, the testbed, as viewed from the 480x640 pixel resolution camera, is decomposed into an 18x25 area based on half the robot's length of 26 pixels. It is worth mentioning that the users can specify the number of grids through the GUI.

**Phase 2.** Grid cells are given one of three initial values: (0) free cell, (-1) cell containing part of an obstacle, and (1) target location.

**Phase 3.** Starting at the target location and iterating out, free cells sharing 8-point connectivity with the cells from the previous iteration are given a value of the current iteration. This propagation continues until the starting cell (where the robot currently is) is reached. Figure 35 shows an example of this propagation.

0	0	-1	-1	4	3	2	2	2	3	0	0	-1	-1	4	3	2	2	2	3
0	0	-1	-1	4	3	2	1	2	3	0	0	-1	-1	4	3	2	1	2	3
0	0	-1	-1	4	3	2	2	2	3	0	0	-1	-1	4	3	2	2	2	3
0	0	-1	-1	4	3	3	3	3	3	0	0	-1	-1	4	3	3	3	3	3
0	0	0	5	4	4	4	4	4	4	0	0	6	5	4	4	4	4	4	4
0	0	0	5	5	-1	-1	-1	-1	5	0	0	6	5	5	-1	-1	-1	-1	5
0	0	0	0	0	-1	-1	-1	-1	0	0	0	6	6	6	-1	-1	-1	-1	6
0	0	0	0	0	-1	-1	-1	-1	0	0	0	0	0	0	-1	-1	-1	-1	0
0	0	0	0	0	-1	-1	-1	-1	0	0	0	0	0	0	-1	-1	-1	-1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 35: Iteration 5(top) and iteration 6 (bottom) of an example wavefront path planning.

**Phase 4.** A path is propagated, starting from the robot's current cell, by selecting the lowest positive number sharing 8 point connectivity with the cell. The path generation iterates until the target is reached Figure 36.

0	0	-1	-1	4	3	2	2	2	3
0	0	-1	-1	4	3	2	1	2	3
0	0	-1	-1	4	3	2	2	2	3
0	0	-1	-1	4	3	3	3	3	3
0	0	0	-1	-1	-1	-1	-1	-1	4
0	0	0	-1	-1	-1	-1	-1	-1	5
0	0	0	-1	-1	-1	-1	-1	-1	6
0	15	15	-1	-1	-1	-1	-1	-1	7
0	15	14	-1	-1	-1	-1	-1	-1	8
0	15	14	13	13	13	13	13	9	9

Figure 36: Path found from wavefront path planning

### Testbed Simulations-UCF

More than 100 hardware experiment tests have been conducted in the developed low-cost, vision based testbed, and the following describes a typical testbed run using two robots. The initial positions  $\mathbf{r}_{a,0}$  obtained from the vision system are  $[33.6,140.8]cm$  and  $[35.2,70.2]cm$  for Robot 1 and Robot 2, respectively, and the final targets are  $[248,184]cm$  and  $[248,8]cm$ . The optimal path is generated for each of the robots based on an initial known set of three obstacles with the following center points defined as  $\mathbf{r}_{obs,1} = [86.8,171.2]cm$ ,  $\mathbf{r}_{obs,2} = [113.6,146.0]cm$ , and  $\mathbf{r}_{obs,3} = [146.0,125.6]cm$  (Figure 37). The buffer for all of these obstacles is set as  $a_{buf} = 17.6cm$ .

The inter-robot collision avoidance is considered in Robot 2 to avoid the collision with Robot 1. This collision avoidance behaves as follows. Robot 1 first generates its minimum-time trajectory while taking into account all boundary conditions. Second, Robot 2 then generates its trajectory while taking into account the boundary conditions and the collision avoidance constraint  $\|\mathbf{r}_{a,2} - \mathbf{r}_{a,1}\| \geq d_r$ , where  $d_r = 17.6cm$  is assumed to be the diagonal of a circle that encloses Robot 1.

Once the optimal trajectories are computed using the BVMC method, each robot follows its path. After that in this scenario, two new obstacles appear at  $[178.8,20.2]cm$  and  $[186.8,113.6]cm$ , and the robots will stop and wait for a new path to be generated. Then the robots follow the re-planned path to reach their target destinations as shown in Figure 38.

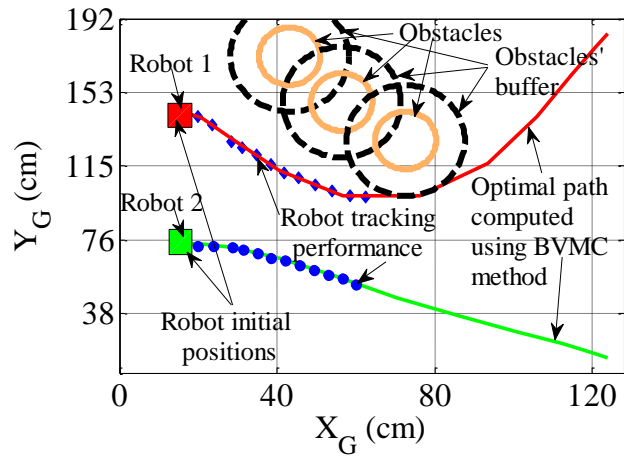


Figure 37: The minimum time trajectory planned considering the known three obstacles in the first section (Case 1)

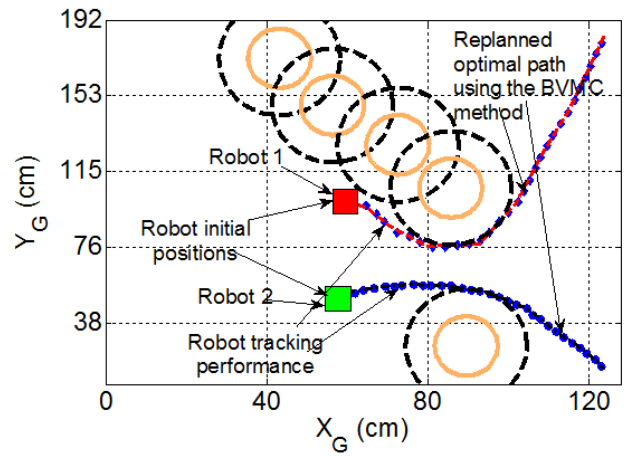


Figure 38: The minimum time trajectory re-planned considering all five obstacles in the second section (Case 1)

Figure 39 shows the combination of the two sections overlaid upon an image of the testbed as seen by the vision system.

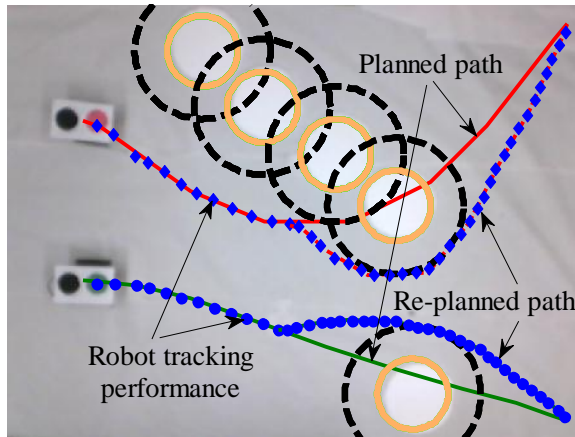


Figure 39: Combination of the first and second sections of a testbed run (Case 1)

The underlying image is the initial setup of the testbed. For this particular run, the optimal paths of Robot 1 are computed using 2.31 seconds after six tries for the first section (S1) and 6.10 seconds after three tries for the second section (S2). Robot 2 took 6.39 seconds to compute the optimal trajectory after five tries for its first section and while its second section took only 2.98 seconds for a single try. The performance indices for Robot 1 and Robot 2 can be seen in Table 3.

Table 3: Testbed Result Case I

	CPU Time (s)		BVMC Tries		Performance Index (s)	
	<b>S1</b>	<b>S2</b>	<b>S1</b>	<b>S2</b>	<b>S1</b>	<b>S2</b>
Robot 1	2.31	6.10	6	3	11.58	8.64
Robot 2	6.39	2.98	5	1	9.81	6.26

Figure 40 and Figure 41 show two additional experiment runs (Case 2 and Case 3), and Tables II and III show their respective computational cost, number of tries, and performance indices. Both runs show that the collision avoidance with both obstacles and other robots is satisfied in the closed quarters.

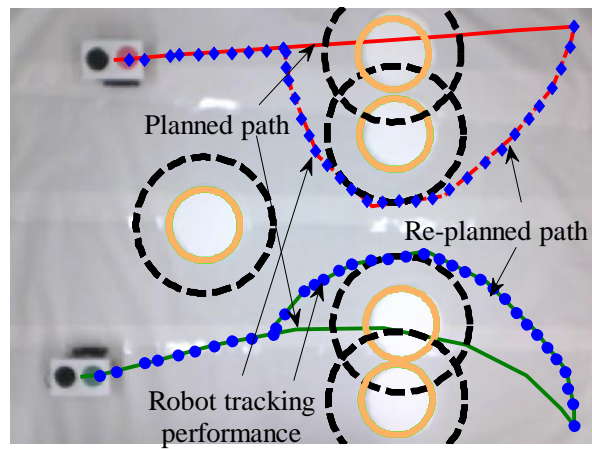


Figure 40: Combination of the first and second sections of a testbed run (Case 2)



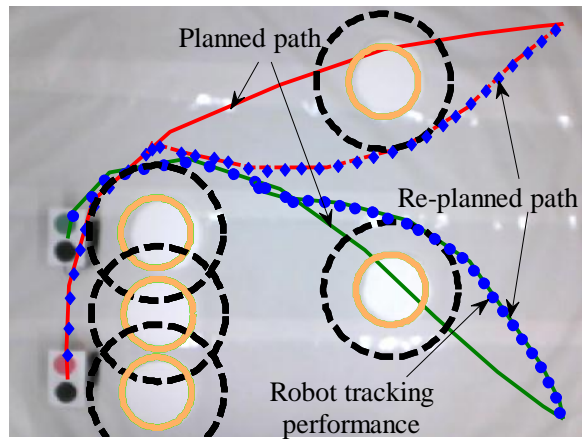


Figure 41: Combination of the first and second sections of a testbed run (Case 3)

As shown in Table 3, Table 4, and

Table 5, the minimum-time trajectories can be computed within the range of 0.82 seconds to 6.39 seconds depends on how many tries it has to take. These testbed runs show the applicability of the BVMC method in a real environment. If the program is coded in C/C++, the computational cost can be further reduced significantly.

Table 4: Testbed Result Case 2

	CPU Time (s)		BVMC Tries		Performance Index (s)	
	<b>S1</b>	<b>S2</b>	<b>S1</b>	<b>S2</b>	<b>S1</b>	<b>S2</b>
Robot 1	0.82	5.82	1	3	8.88	10.27
Robot 2	3.04	6.17	1	2	10.23	8.69

Table 5: Testbed Result Case 3

	CPU Time (s)		BVMC Tries		Performance Index (s)	
	<b>S1</b>	<b>S2</b>	<b>S1</b>	<b>S2</b>	<b>S1</b>	<b>S2</b>
Robot 1	1.24	1.56	1	1	13.94	9.59
Robot 2	1.88	1.61	1	1	12.28	8.79

Tesbed Simulations –AFRL

***Test Scenario 1***

This scenario demonstrates the hardware and algorithm’s ability to navigate a robot through a “changing” narrow corridor of obstacles. As seen in Figure 42(*left*), the robot starting at the pixel location of [92,142], only has the knowledge of obstacles along the first half of the testbed. The optimal path based on this has a performance index of 33.17 seconds. Stopping at the pixel location of [179,215], the remaining obstacle locations are made available as shown in Figure 42(*right*). This new path has a performance index of 28.74 seconds.

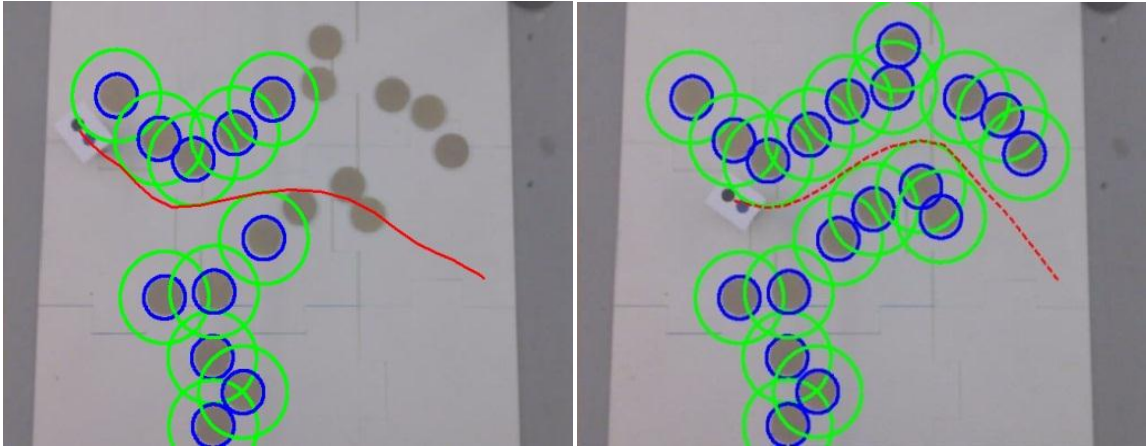


Figure 42: Test Scenario 1 (*left*) Section 1, (*right*) Section 2

***Test Scenario 2***

Two robots, starting at pixel locations [170,130] and [111,334], are given the knowledge of the first half of the environment with desired goals of [540,200] pixels and [540,300] pixels, respectively (Figure 43). The path for robot 1 has a performance index of 29.99 seconds, while the robot 2's performance index is 29.14 seconds. Upon reaching the pixel locations of [195,165] and [183,316], the rest of the obstacle locations are revealed and robots replan their trajectories. In this second planning, the path for robot 1 has a performance index of 23.17 seconds while robot 2's performance index is 23.17 seconds.

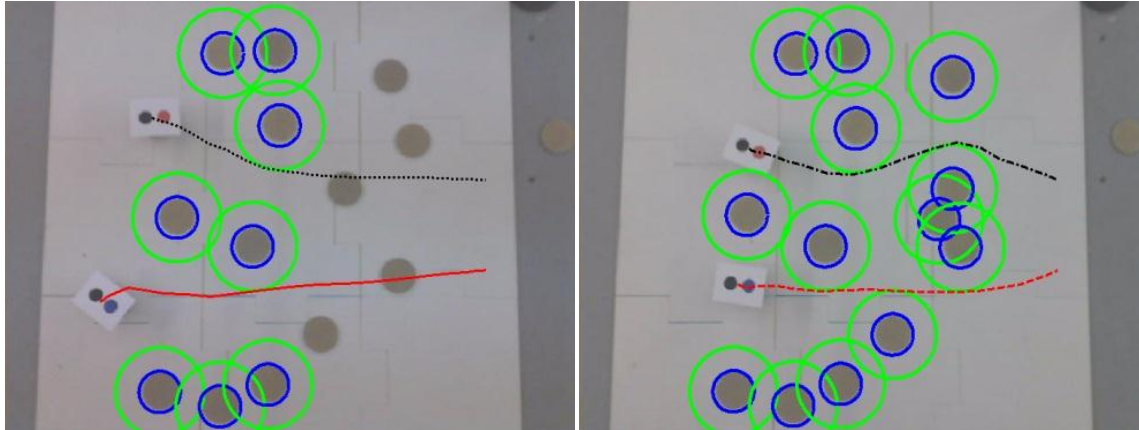


Figure 43: Test Scenario 2 (*left*) Section 1, (*right*) Section 2

### ***Test Scenario 3***

Three robots, starting at the pixel locations of [488,426], [92,349], and [106,92], are given the full knowledge of their environment (Figure 44). The achieved optimal performance indices are 29.30 seconds, 40.55 seconds, and 38.41 seconds, respectively. Stopping at the pixel locations of [386,433], [133,234], and [217,174], robot trajectories are replanned. This replanning resulted in the performance indexes of 22.85 seconds, 30.28 seconds, and 27.82 seconds for robots 1, 2 and 3 respectively.

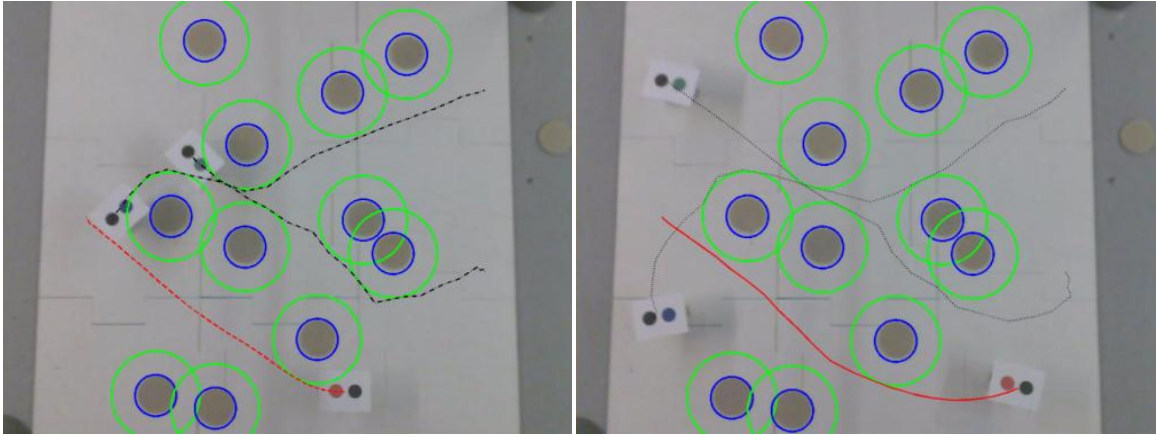


Figure 44: Test Scenario 3 (*left*) Section 1, (*right*) Section 2

## CHAPTER FIVE: CONCLUSIONS AND FUTURE WORK

### Conclusions

The basics of camera modeling, distortion compensation, stereoscopic reconstruction and calibration have been covered. Implementing these, two very different vision-based testbeds were developed.

The first was designed to test and control shape memory polymers, a kind of smart material capable of holding a deformed shape until an external stimulus excites it back to its original shape. In this testbed points of interest on samples were tracked using uniquely colored dots and RGB thresholding. A PID based controller was shown to repeatedly control deflection angles to a desired position. Additionally, input and output data was used to create a state-space model of a sample that could reproduce deflection angle response to varying voltage input in simulation.

A mobile robot testbed was the second vision-based application presented. Robots were located and obstacles were detected by creating normalized cross correlations using templates of each. An easy to use interface was designed to support a dynamic number of robot connections, automatic calibration of parameters used in path tracking, and quickly define and run experiments. The recently developed, bio-inspired, trajectory optimization method BVMC was shown to work in a physical environment and then improved by using the wavefront path planning method as an initial guess for the optimization. This enhanced BVMC was shown to quickly find optimal trajectories in obstacle dense environments that robots could then track.

### Future Work

Vision based testbeds are developed for two interesting control experiments. Below are a small number of suggestions to improve the capabilities of the testbeds.

#### *SMP Testbed*

**Pneumatic artificial rubber muscle.** Or something similar to the work seen in [94]. It could be easily modified for 3D motion in a hemisphere and could play a large role as a pointing actuator. This would also give a reason to make use of the stereoscopic reconstruction capability of the testbed.

**Portable testbed.** Currently, the testbed is in what is fondly known as a “breadboard” stage. It would aesthetically benefit from light controlled box that housed the vision system, test area, and sensors. This would also reduce the amount of calibration require to account for location and lighting changes.

#### *Mobile Robot Testbed*

**Decentralized path tracking.** Currently, the robots in the testbed are little more than puppets receiving direct commands from the laptop serially. While this was fine for the applications shown here, this would be a closer approximation to real-world applications and would free up processing power on the laptop for other things. Moving control schemes to the robots would also benefit from onboard sensors: dead-reckoning and other sensor based method of position estimation could be implemented for faster, more accurate trajectory following and robot/obstacle detection.

**Wi-Fi based communication.** The disadvantage of the Bluetooth based communication scheme is that it will support a maximum of seven simultaneous connections. To examine

systems with a larger number of robots, another method of communication must be implemented. The simplest and most universal solution is Wi-Fi which can also be easily encrypted.

**Quadrotor Platform.** Adding quadrotor platforms would add a third dimension for trajectory considerations and significantly increase the simulation capabilities. This could even replace the overhead vision systems with mobile vision systems able to focus on certain areas/objects of interest or provide an overall view of the test area.



## LIST OF REFERENCES

- [1] E. L. Hall, "Measuring curved surfaces for robot vision," *Computer*, vol. 15, pp. 42-54, 1982.
- [2] R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision," *ACM Computing Surveys*, vol. 18, pp. 67-108, 1986.
- [3] A. Prati, R. Vezzani, L. Benini, and E. Farella, "An integrated multi-modal sensor network for video surveillance," in *The third ACM International Workshop on Video Surveillance & Sensor Networks*, New York, NY, 2005, pp. 95-102.
- [4] H. Buxton and S. Gong, "Visual surveillance in a dynamic and uncertain world," *Artificial Intelligence*, vol. 78, pp. 431-459, 1995.
- [5] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 34, pp. 334-352, 2004.
- [6] F. Pedreschi, J. Leon, D. Mery, and P. Moyano, "Development of a computer vision system to measure the color of potato chips," *Food Research International*, vol. 39, pp. 1092-1098, 2006.
- [7] J. Jurkovic, M. Korosec, and J. Kopac, "New approach in tool wear measuring technique using CCD vision system," *International Journal of Machine Tools and Manufacture*, vol. 45, pp. 1023-1030, 2005.
- [8] J. Blasco, N. Aleixos, and E. Molto, "Machine vision system for automatic quality grading of fruit," *Biosystems Engineering*, vol. 85, pp. 415-423, 2003.

- [9] V. Carbone, M. Carocci, E. Savio, G. Sansoni, and L. D. Chiffre, "Combination of a vision system and a coordinate measuring machine for the reverse engineering of freeform surfaces," *The international Journal of Advanced Manufacturing Technology*, vol. 17, pp. 263-271, 2001.
- [10] S. Soatto, R. Frezza, and P. Perona, "Motion estimation via dynamic vision," *IEEE Transactions on Automatic Control*, vol. 41, pp. 393-413, 1996.
- [11] S. Se, H.-K. Ng, P. Jasiobedzki, and T.-J. Moyung, "Vision based modeling and localization for planetary exploration rovers," presented at the 55th International Astronautical Congress, Vancouver, Canada, 2004.
- [12] K. M. Lum and Y. D. Wong, "A before-and-after study of driver stopping propensity at red light camera intersections," *Accident Analysis & Prevention*, vol. 35, pp. 111-120, 2003.
- [13] C. Bregler, C. Castiglia, J. DeVincezo, R. L. DuBois, K. Feeley, T. Igoe, *et al.*, "Squidball: an experiment in large-scale motion capture and game design," *Intelligent technologies for interactive entertainment*, vol. 3814, pp. 23-33, 2005.
- [14] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande, "Turning to the masters: motion capturing cartoons," in *29th annual conference on Computer graphics and interactive techniques*, New York, NY, 2002, pp. 399-407.
- [15] A. Lendlein, H. Jiang, O. Junger, and R. Langer, "Light-induced shape-memory polymers," *Nature*, pp. 879-882, 2005.
- [16] A. Lendlein and S. Kelch, "Shape-memory polymers," *Chem. Int. Ed.*, pp. 2034-2057, 2002.

- [17] J. S. Leng, H. B. Lu, Y. J. Liu, W. M. Huang, and S. Y. Du, "Shape-memory polymers: a class of novel smart material," *MRS Bulletin*, pp. 848-855, 2009.
- [18] D. J. Maitland, M. F. Metzger, D. Schumann, A. Lee, and T. S. Wilson, "Laser activated shape memory polymer microactuator for treating ischemic stroke," *Lasers Surg Med*, pp. 1-11, 2002.
- [19] I. Bellin, S. Kelch, R. Langer, and A. Lendlein, "Polymeric triple shape materials," in *Natl Acad. Sci.*, USA, 2006, pp. 18043-18047.
- [20] R. Vaia, "Nanocomposites-remote-controlled actuators," *Nature Mater.*, pp. 429-430, 2005.
- [21] N. G. Sahoo, Y. C. Jung, N. S. Goo, and J. W. Cho, "Conducting shape memory polyurethane-polypyrrole composites for an electroactive actuator," *Macromolr Mater Eng.*, pp. 1049-1055, 2005.
- [22] A. M. Schmidt, "Electromagnetic activation of shape memory polymer networks containing magnetic nanoparticles," *Macromol. Rapid Commun.*, pp. 1168-1172, 2006.
- [23] M. Y. Razzaq, M. Anhalt, L. Frommann, and B. Weidenfeller, "Mechanical spectroscopy of magnetite filled polyurethane shape memory polymers," *Materials, Science & Engineering*, vol. 471, pp. 57-62, 2007.
- [24] H. B. Lu, J. S. Leng, Y. J. Liu, and S. Y. Du, "Shape-memory polymer in response to solution," *Adv. Eng. Mater.*, vol. 10, pp. 592-595, 2008.
- [25] H. B. Lu, Y. J. Liu, J. S. Leng, and S. Y. Du, "Qualitative separation of the effect of solubility parameter on the recovery behavior of shape-memory polymer," *Smart Mater. Struct.*, vol. 18, pp. 1-5, 2009.

- [26] J. Leng, "Comment on water-driven programable polyurethane shape memory polymer: demonstration and mechanism," *Applied Physics Letters*, vol. 92, pp. 206105-206105-2, 2008.
- [27] J. Gou, "Single-walled nanotube bucky paper and nanocomposite," *Polymer International*, vol. 55, pp. 1283-1288, 2006.
- [28] J. Gou, S. O'Brain, H. Gu, and G. Song, "Damping augmentation of nanocomposites using carbon nanofiber paper," *Nanomaterials*, vol. 2006, pp. 1-7, 2006.
- [29] A. Lendlein and R. Langer, "Biodegradable, Elastic Shape-Memory Polymers for Potential Biomedical Applications," *Science*, vol. 296, pp. 1673-1676, 2002.
- [30] R. Langer and D. A. Tirrell, "Designing materials for biology and medicine," *Nature*, vol. 428, pp. 487-492, 2004.
- [31] A. L. Browne and N. L. Johnson, "Method for controlling airflow," United States of America Patent, 2007.
- [32] X. Lan, Y. Liu, H. Lv, X. Wang, J. Leng, and S. Du, "Fiber reinforced shape-memory polymer composite and its application in a deployable hinge," *Smart Mater. Struct.*, vol. 18, pp. 1-6, 2009.
- [33] S. Kelch and A. Lendlein, "Shape memory polymers," *Angew. Chem Int. Edn. Engl.*, vol. 41, pp. 2034-2057, 2002.
- [34] S. Berman and Y. Edan, "Decentralized autonomous AGV system for material handling," *International Journal of Production Research*, vol. 40, pp. 3995-4006, 2002.
- [35] K. Wurm, C. Dornhege, P. Eyerich, C. Stachniss, B. Nebel, and W. Burgard, "Coordinated exploration with marsupial teams of robots using temporal symbolic

- planning," presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 2010.
- [36] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," *Lecture Notes in Computer Science*, vol. 5515, pp. 117-139, 2009.
- [37] M. Takahashi, T. Suzuki, H. Shitamoto, T. Moriguchi, and K. Yoshida, "Developing a mobile robot for transport applications in the hospital domain," *Robotics and Autonomous Systems*, vol. 58, pp. 889-899, 2010.
- [38] Y. K. Hwang and N. Ahuia, "Gross motion planning," *ACM Computing Surveys*, vol. 24, pp. 219-291, 1992.
- [39] E. Masehian and D. Sedighizade, "Classic and heuristic approaches in robot motion planning," in *World Academy of Science, Engineering and Technology*, Germany, 2007, pp. 101-106.
- [40] S. Keshmiri and S. Pavandeh, "An overview of mobile robotic agents' motion planning in dynamic environments," in *14th LASTED International Conference*, Cambridge, MA, 2009, pp. 152-159.
- [41] V. Kunchev, L. Jain, V. Ivencevic, and A. Finn, "Path planning and obstacle avoidance for autonomous robots: a review," *Lecture Notes in Computer Science*, vol. 4252, pp. 537-544, 2006.
- [42] T. C. Hu, A. B. Kahang, and G. Robins, "Optimal robust path planning in general environments," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 775-784, 1993.

- [43] G. E. Jan, K. Y. Chang, and I. Parberry, "Optimal path planning for mobile robot navigation," *IEEE/ASME Transactions on Mechatronics*, vol. 13, pp. 451-460, 2008.
- [44] N. Rowe and R. Richbourg, "An efficient snell's law method for optimal path planning across multiple two-dimensional, irregular, homogenous-cost regions," *The International Journal of Robotics Research*, vol. 9, pp. 46-66, 1990.
- [45] Z. Qu, J. Wang, and C. E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presense of moving obstacles," *IEEE Transactions on Robotics*, vol. 20, pp. 978-993, 2004.
- [46] D. Alves, B. O. Vaz, R. S. Inoue, and V. Grassi, "Kinomatic motion planning of a skid-steering mobile robot using RRTs," presented at the Latin American Robotics Symposium and Intelligent Robotic Meeting, San Bernardo de Campo, Brazil, 2010.
- [47] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, pp. 933-945, 2009.
- [48] Y. Xu, "Motion camouflage and constrained suboptimal trajectory control," presented at the 2007 AIAA Guidance, Control and Dynamics Conference, Hilton Head, SC, 2007.
- [49] Y. Xu and G. Basset, "Sequential virtual motion camouflage method for nonlinear constrained optimal trajectory control," *Automatica*, vol. 48, pp. 1273-1285, 2012.
- [50] G. Basset, Y. Xu, and N. Li, "Fast trajectory planning via the b-spline augmented virtual motion camouflage approach," presented at the 50th IEEE Conference on Decision and Control, Orlando, FL, 2011.

- [51] T. W. McLain and R. W. Beard, "Unmanned air vehicle testbed for cooperative control experiments," in *American Control Conf.*, Boston, Massachusetts, 2004, pp. 5327-5331.
- [52] D. Johnson, T. Stack, R. Fish, D. M. Flickinger, L. Stoller, R. Ricci, *et al.*, "Mobile emulab: a robotic wireless and sensor network testbed," in *25th Conf. on Computer Communications*, 2006, pp. 1-12.
- [53] Z. Wang, F. Gu, Y. He, J. Han, and Y. Wang, "Design and implementation of multiple-rotorcraft-flying-robot testbed," presented at the IEEE Int. Conf. on Robotics and Biomimetics, Phuket, Thailand, 2011.
- [54] M. Barbosa, A. Bernardino, D. Figueira, J. Gaspar, N. Goncalves, P. U. Lima, *et al.*, "ISRobotNet: A Testbed for sensor and robot network systems," presented at the IEEE Int. Conf. on Intelligent Robots and Systems, St. Louis, 2009.
- [55] L. E. Parker, B. Birch, and C. Reardon, "Indoor target intercept using an acoustic sensor network and dual wavefront path planning," in *IEEE Int. Conf. Intelligent Robotic Systems*, 2003, pp. 278-283.
- [56] F. Liang, R. Sivilli, J. Gou, and Y. Xu, "Electrical Actuation and Vision-Based Control of Shape Memory Polymer Carbon Nanocomposites," in *56th Society for Advancement of Material and Process Engineering Symposium and Exhibition*, Baltimore, Maryland, 2012.
- [57] F. Liang, R. Sivilli, J. Gou, and Y. Xu, "Electrical actuation and precisely control of carbon nanofiber paper based shape memory polymer nanocomposites " *Smart Materials and Structures*, 2012.

- [58] F. Liang, R. Sivilli, Y. Xu, and J. Gou, "Shape Memory Polymer Nanocomposites for High Speed Electrical Actuation," in *55th Society for Advancement of Material and Process Engineering Symposium and Exhibition*, Long Beach, CA, 2011.
- [59] R. Sivilli, Y. Xu, and K. Pham. (2012, Pathfinding for mobile sensor networks on the fly. *SPIE Newsroom*.
- [60] G. Basset, R. Sivilli, Y. Xu, and K. Pham, "Minimum-time obstacle avoidance trajectory planning for vision based robots," *Robotica*, 2012.
- [61] R. Sivilli, Y. Xu, and K. Pham, "A vision based robot testbed for single or multiple vehicles' trajectory planning," presented at the American Control Conference, Washington, DC, 2013.
- [62] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1330-1334, 2000.
- [63] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 965-980, 1992.
- [64] V. Douskos, I. Kalisperakis, and G. Karra, "Automatic calibration of digital cameras using planar chess-board patterns," presented at the 8th Conf. Opt. 3-D Measurement Techniques, Wichmann, 2007.
- [65] D. C. Brown, "Close-range camera calibration," *Photogrammetric Engineering* pp. 855-866, 1971.
- [66] W. Faig, "Calibration of close-range photogrammetric systems: Mathematical formulation," *Photogrammetric Eng. Remote Sensing*, vol. 41, pp. 1479-1486, 1975.



- [67] J. C. McGlone, E. M. Mikhail, J. S. Bethel, and R. Mullen, *Manual of Photogrammetry*, 5th ed.: ASPRS, 2004.
- [68] S. S. Rao, *Applied numerical methods for engineers and scientists*. Upper Saddle River, NJ 07458: Prentice Hall, 2002.
- [69] G. Bradski and A. Kaehler, *Learning OpenCV*, 1st ed. Sebastopol, CA, 95472: O'Reilly Media, Inc., 2008.
- [70] V. Douskos, I. Kalisperakis, and G. Karras, "Automatic calibration of digital cameras using planar chess-board patterns," in *8th Conference on Optical 3-D Measurement Techniques*, Zurich, Switzerland, 2007, pp. 132-140.
- [71] H. Hirshmuller, "Real-time map building from a stereo camera under unconstrained 3d motion," in *Faculty Research Conference*, Leicester, UK, 2003, pp. 1-6.
- [72] M. A. Garcia and A. Solanas, "3D simultaneous localization modeling from stereo vision," in *IEEE International Conference on Robotics & Automation*, Tarragona, Spain, 2004, pp. 847-853.
- [73] P. Liu, A. Willis, and Y. Sui, "Stereoscopic 3D reconstruction using motorized zoom lenses within an embedded system," *Image Processing: Machine Vision Applications II*, vol. 7251, pp. 1-12, 2009.
- [74] A. Fusiello, E. Trucco, and A. Verri, "Rectification with unconstrained stereo geometry," in *Proceedings of the British Machine Vision Conference*, 1997, pp. 400-409.
- [75] J. H. Lumkes, *Control Strategies for Dynamic Systems: Design and Implementation*. New York, NY: Marcel Dekker, 2001.

- [76] M. Jun and M. G. Safonov, "Automatic PID tuning: an application of unfalsified control," in *International Symposium on Computer Aided Control System Design*, Kohala Cost, Hawaii, 1999, pp. 328-333.
- [77] J. Lee, "On methods of improving performance of PI-type fuzzy logic controllers," *IEEE Transactions on Fuzzy Systems*, vol. 1, pp. 298-301, 1993.
- [78] A. Hansson, P. Gruber, and J. Todtil, "Fuzzy anti-reset windup for PID controllers," *Control Engineering Practice*, vol. 2, pp. 389-396, 1994.
- [79] F. P. Incropera, D. P. Dewitt, T. L. Bergman, and A. S. Lavine, *Fundamentals of Heat and Mass Transfer*, 6th ed.: John Wiley & Sons, 2007.
- [80] W. L. Brogan, "Modern Control Theory," in *Modern Control Theory*, P. Fraccio and B. M. d. Leon, Eds., ed Upper Saddle River, NJ: Prentice Hall, 1991, pp. 507-541.
- [81] D. S. Naidu, "Linear Quadratic Optimal Control Systems I," in *Optimal Control Systems*, ed Boca Raton, Florida: CRC Press LLC, 2003, pp. 101-150.
- [82] D. J. Hartl and D. C. Lagoudas, "A constitutive theory for shape memory polymers. part II: a linearized model for small deformations," *Journal of the Mechanics and Physics of Solids*, vol. 56, pp. 1766-1778, 2009.
- [83] T. D. Nguyen, H. J. Qi, F. Castro, and K. N. Long, "A thermoviscoelastic model for amorphous shape memory polymers: incorporating structural and stress relaxation," *Journal of the Mechanics and Physics of Solids*, 2008.
- [84] J. Diani, Y. Liu, and K. Gall, "Finite strain 3d thermoviscoelastic constitutive model for shape memory polymers," *Polymer Engineering Science*, vol. 46, pp. 1-7, 2006.

- [85] M. Baghani, R. Naghdabadi, J. Arghavani, and S. Sohrabpour, "A thermodynamically-consistent 3D constitutive model for shape memory polymers," *International Journal of Plasticity*, vol. 35, pp. 13-30, 2012.
- [86] P. Ghosh, "Thermomechanical modeling of a shape memory polymer," M.S.M.E, Mechanical Engineering, Texas A&M, 2008.
- [87] E. Azamasab and X. Hu, "An integrated multi-robot test bed to support incremental simulation-based design," presented at the 2007 IEEE International Conference on System of Systems Engineering, 2007.
- [88] T. A. Riggs, T. Inanc, and W. Zhang, "An autonomous mobile robotics testbed: construction, validation, and experiments," *IEEE Transactions on Control Systems Technology*, vol. 18, pp. 757-766, 2010.
- [89] N. Michael, J. Fink, and V. Kumar, "Experimental testbed for large multi-robot teams," *IEEE Robotics & Automation Magazine*, vol. 15, pp. 53-61, 2010.
- [90] J. P. Laumond, S. Sekhavat, and F. Lamiroux, *Robot motion planning and control*. LASS-CNRS, Toulouse: LAAS, 1998.
- [91] K. Park, H. Chung, J. Choi, and J. G. Lee, "Dead reckoning navigation for an autonomous mobile robot using a differential encoder and a gyroscope," presented at the ICAR '97, Monterey, CA, 1997.
- [92] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, pp. 193-207, 1998.

- [93] L. E. Parker, B. Birch, and C. Reardon, "Indoor target intercept using an acoustic sensor network and dual wavefront path planning," in *IEEE International Conference on Intelligent Robots and Systems*, St Louis, 2003, pp. 2827-2833.
- [94] K. Takashima, T. Noritsugu, J. Rossiter, S. Guo, and T. Mukai, "Development of curved type pneumatic artificial rubber muscle using shape-memory polymer," presented at the SICE Annual Conference, Tokyo, Japan, 2011.