A Testbed Architecture for MAGICC Applications

Brett J. Young   Randal W. Beard
Electrical and Computer Engineering Department
Brigham Young University

Abstract

A testbed is developed for testing Multi-AGent Intelligent Coordination and Control (MAGICC). The robots act as agents which make decisions and act upon them via classical control theory. The testbed consists of sensors(camera and encoders), embedded controllers, modems and a computer network which are integrated into a hierarchical control which has decision makers, controllers, estimators and predictors. The testbed has successfully been applied to initializing the robots into geometric formation. Simulation and Hardware results are presented.

1 Introduction

A fascinating problem emerging in engineering is that of having multiple agents working together as a group to achieve some goal or mission. Examples of multiple agents include satellites flying in formation as well as robots working cooperatively to achieve some planetary exploration. The mathematics of having satellites move in formation or robots cooperate traditionally falls under game theory. The problem becomes complex with more than two players and the payoff matrices can literally take hours to calculate [Randal Beard, 1998] Also, conditional constraints may be imposed on the problem such as limited sensor range and communication. Having global coordination in such situations may be difficult. However, humans work in such environments everyday. The following list specifies some of the expectations for multiple coordinating agents:

Independence– Each agent should be able to assess the situation and act upon what data it has available, especially when communication is limited.

Robustness– As given in [Brooks, 1986], the agents should act rational despite altered constraints.

Stability– Where applicable control theory should be applied. For example, if a robot decides to go to point A from point B, it should be able to guarantee convergence to its decision.

Achievability– The agents should be able to guarantee some measure of success with respect to their group goals. For example, formation maneuvers should have guaranteed success.

The multi-agent problem has started to receive some attention in robotics [Brooks, 1986]. purposes a layered architecture which has low level controls as well a heuristic decision maker. Another approach used specifically for formation control relies completely on heuristics [Balch and Arkin, 1998], where the robots attempt to imitate flocking birds. Such *ad hoc* approached work reasonably well for specific applications. Yet, they often fail in general and new heuristics must be developed. Also the mathematical analysis of heuristics do not develop issues such as stability and convergence.

Recently, a new tool for controlling multiple agent has been developed at BYU [Stirling and Goodrich, 1999]. The theory uses a mathematical tool for decision making based on Isaac Levi's ideas on rational decision making. The advantage is that each agent can make coordinating decisions independently. This is different from other schemes since each agent infers what the actions of his neighboring agents. They relate to each other through an interdependence function. This approach does not try to find the "best solution" or minimize a pay-off matrix. Instead it tries to find good solutions with respect to some criteria (e.g. having robots all go to different formation targets in a reasonable amount of time). The remainder of this paper outlines a hardware architecture which has been constructed to implement and test

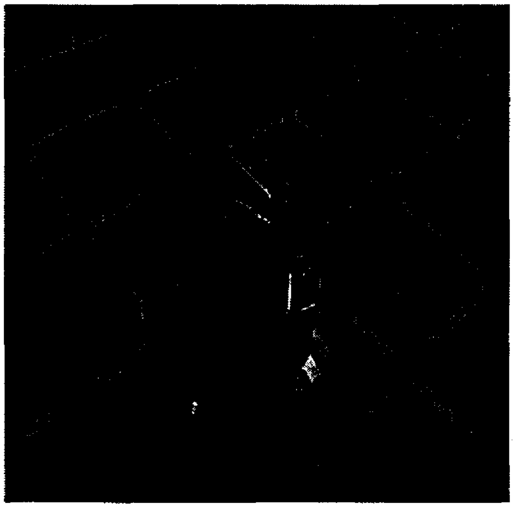

Figure 1: The robots built for the MAGICC LAB

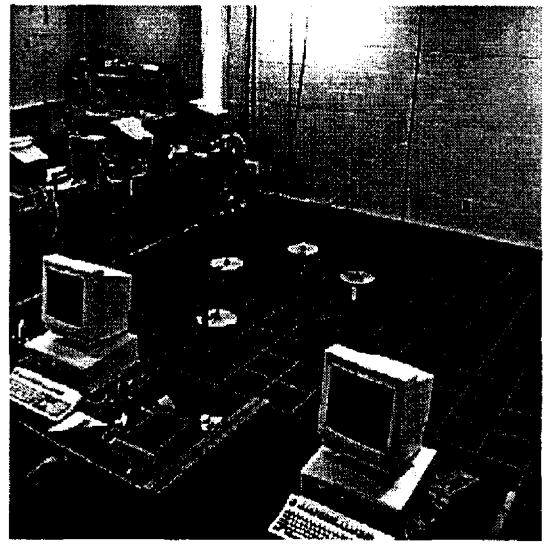

Figure 2: The robots and their playing field

multiple agent coordinated control via multiple wheeled robots. The overall structure developed for the testbed will be presented. To demonstrate the functionality of the testbed, simulation and experimental results will illustrate our solution to the problem of regulating a robot to a specified location

2 Basic Hardware

First the robots had to be designed and built physically. The design agreed upon was a two wheeled mobile robots which had two casters: one in the front and one in the back. The two main wheels are driven by Pitman motors. The Pitman motors also have encoders mounted to them. We built our own decoder board to read the decoder chips. The Handy Board developed at M.I.T. was used as an embedded controller on the robot [Martin, 1998]. Encoder counts were sent to a computer via a radio modem, at an update rate of 0.2 samples/second. A camera was mounted above the testbed to provide global sensing capabilities. Colored dots were placed on top of the robots so that the camera could spot the robots via an edge detection algorithm. Also an algorithm was developed and used to undo the distortion via edge detection. A picture of one robots can be seen in Figure 1. The picture of the testbed can be seen in Figure 2.

3 Robot Model

The governing equations of motion for the robot are important. A two wheeled mobile robot is a *non-holonomic system* which means that the velocity of the robot is constrained to be in its direction of motion. Let x,y denote the Cartesian coordinates of the robot, θ represents the orientation of the robot, v the linear velocity in the direction of motion and ω the angular velocity of the robot. Thus the position vector of the robots is given by $\vec{x}_{pos} = (x, y, \theta)$ This can be seen in Figure 3. The full state vector is given by $\vec{x} = (x, y, \theta, v, \omega)$

After applying some basic physics to the problem the equations are of the form:
Kinematic equations:

$$\dot{x} = vcos(\theta), \quad (1)$$

$$\dot{y} = vsin(\theta) \quad (2)$$

$$\dot{\theta} = \omega \quad (3)$$

Dynamic equations:

$$m\dot{v} = F \quad (4)$$

$$J\dot{\omega} = \tau, \quad (5)$$

where m is the mass, J the inertia, F is force and τ the torque. Thus the equations are separable into linear and rotational motion, a fact

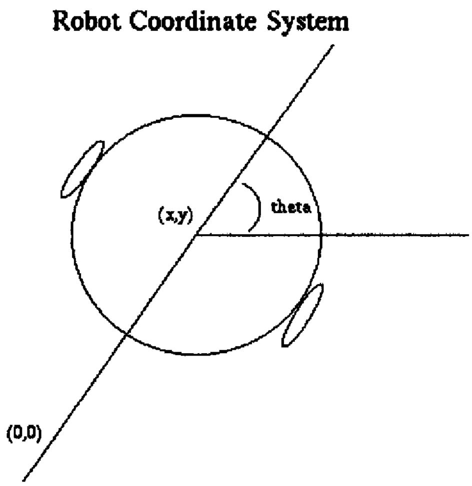

Figure 3: The generalized position coordinate

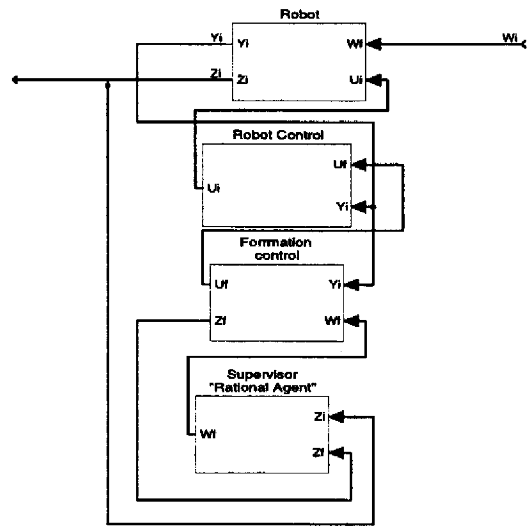

Figure 4: The hierarchy structure

which shall be exploited in the control. The kinematic equations are the non-holonomic equations for multiple robots and can be found in [Kolmanovsky and McClamroch, 1995]. The linear and rotational forces must be translated into a force on the wheels. The equation which decomposes these forces into wheel forces is given by:

$$f_{wheel} = \frac{F}{2} - \frac{\tau}{(r_{robot})2} \tag{6}$$

The wheel forces are then further converted into Volts for motor voltage, given by

$$V_{in} = kmp1(f_{wheel}) + kmp2(v_{wheel}) \tag{7}$$

where kmp1,kmp2 are motor constants, and v_{wheel} is the velocity of the wheel. However, the robots were controlled by an embedded controller which uses pulse width modulation and accepts discrete values between -100 to 100, where 100 corresponds to 12V. A look-up table was used to convert from Volts to motor command values. This table is non-linear and was developed by empirical testing.

4 Control Architecture

All of the components in the hardware combine to make a hierarchical structure for the robots. Figure 4 presents this overall structure. The agent or rational decision maker decides where to go. The formation control generates the desired trajectories. Note that for regulation these are just set points and orientation. The desired values are fed into the low-level control which produces motor commands to the robots.

5 Estimator/Predictor

The overall structure of the sensors is similar to that of missiles which use GPS(slow) and accelerometers(fast) to determine position and acceleration. In our case, the camera acts like a GPS(system) getting updates roughly every 2 sec and the encoders measure velocity and are analogous to accelerometers with updates every 0.2 sec. One difference is that the encoders only measure counts of wheel rotations. This is equivalent to measuring changes in x,y, and θ. So, the observer is not just a simple integrator. The velocity and

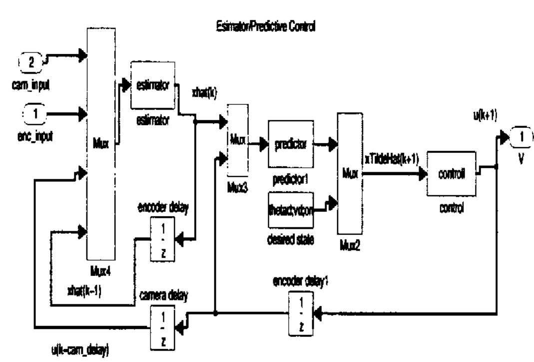

Figure 5: Block Digram of Predictor/Estimator.

angular velocity were estimated from propagating

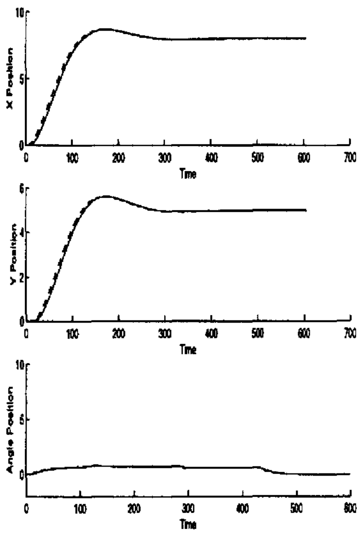

Figure 6: Block Digram of Predictor/Estimator.

a model of the robot using a Runge-Kutta technique and updates to position and angle were a weighted sum of the encoder data and the observer model. The gains used to weight encoder and model data are equivalent to Kalman Gains. When camera updates are received they are 2 sec old, so the old encoder data is weighted with the camera data and propagated thorough a Kalman predictor to update the estimate. The estimator then feeds into a predictor to predict the current state $\vec{x}$ needed in the control. The reason a predictor is necessary is because the control is delayed 0.2 sec by the radio modems. Figure 5 shows a block diagram of the Estimator/ Predictor control. Figure 6 shows how the predictor matched up to robot states in simulating robot regulation.

6 The control

The *non-holonomic* nature of the robot makes the control difficult. Simple PD control will not work due to Brockett's Theorem which states that it is impossible to regulate a *non-holonomic* system with a Continuous Time Invariant Control [Brockett, 1983]. If simple PD control is applied the robots gets stuck in a limit cycle about the desired position and just spins about in a circle around the desired position. Finally a discontinuous control was implemented which put the discontinuity in the angle. The control is similar to that of [Astolfi, 1995]. The basic control Law used was:

Linear Control:

$$m\dot{v} = -k1(x - x_d) - k2v \qquad (8)$$

Rotational Control:

$$J\dot{\omega} = -k3(\theta - \theta_d) - k4\omega, \theta_d = tan^{-1}(\frac{y_d}{x_d}) \qquad (9)$$

This control is given without proof. However, simulation and hardware results are present below in Figure 7. The x,y position converge rapidly. However there is a difference between desired steady angle. One reason is that the robot has static friction which makes it hard to rotate at low velocities. The forces to the motors become low, making it hard for the robot to reach its desired orientation. This is because the gains on the linear and rotational control cannot be simultaneously high or the asymptotic convergence is long and often takes bizarre paths. So when the robot regulates to the right position the rotational control is low and the robot has difficulty overcoming frictional effects. Thinking about what the control is doing, it can be thought of as trying to park your car while driving to the destination. The robot is trying to simultaneously drive the x,y and θ to some desired state. An alternative approach is to have the robot orient itself in the direction it wants to go, move there and re-orient. This has the added advantage that the rotational and linear gains can separately be made large–helping to eliminate problems with friction. This is the basic idea behind the next control presented.

7 A Hybrid Control

The hybrid control transitions between different control laws via a state machine which reaches a final state[Branicky, 1998]. In this case the final state is $\| \tilde{x} \| \leq \epsilon_1, \| \tilde{y} \| \leq \epsilon_2, \| \tilde{\theta} \| \leq \epsilon_3$. Simulation results are presented in Figure 8.

With a good control and estimator in place the testbed can be used for multi-agent coordination problems. The next step in application will be in having the robots do simple formation maneuvers such as expansions and contractions. The low level part of the testbed is in place. Now the decision making and supervisor control levels will be implemented.

8 Conclusion

In this paper a testbed for studying Multi-Agent problems was presented. The testbed was built mainly out of off the shelf components and for a modest price. With the generalized framework in

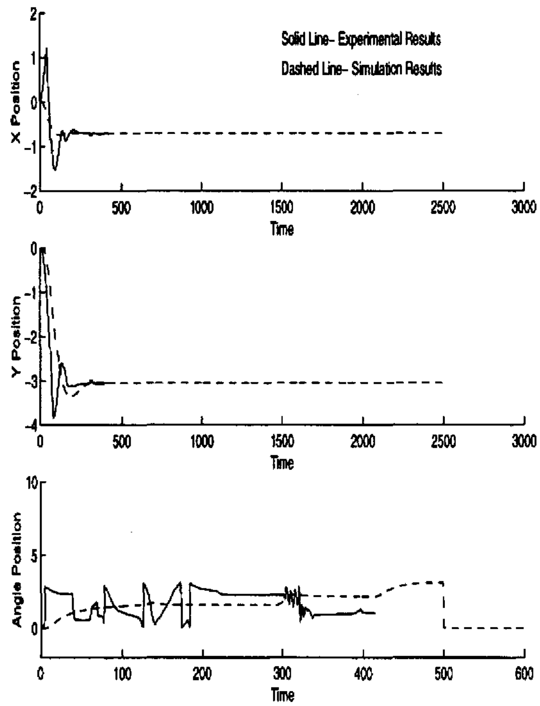

Figure 7: Simulation vs Hardware results for regulation

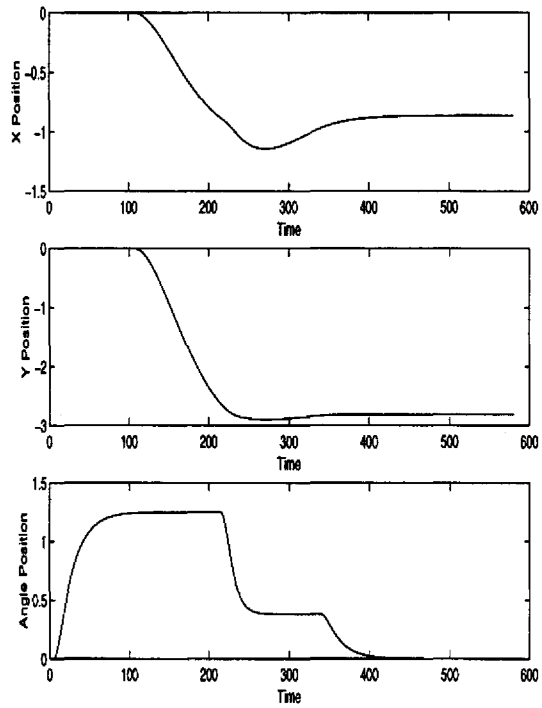

Figure 8: Simulation of a Hybrid Control for regulation.

place, the MAGICC lab is ready to test algorithms for the multi-agent problem. The framework is significant in that it will allow mathematical analysis of stability and achievability while being robust and flexible. The MAGICC lab hopes to create general tools for multi-agent problems and use the testbed to demonstrate some specific applications; namely, formation control and material handling.

References

[Astolfi, 1995] Astolfi, A. (1995). Exponential stabilization of a wheeled mobile robot via discontinous control. *Nonlinear Control System Design Symposium*, pages 741–746.

[Balch and Arkin, 1998] Balch, T. and Arkin, R. C. (1998). Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotices and Automation*, 14 No 6:926–939.

[Branicky, 1998] Branicky, M. S. (1998). Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43 No4:475–482.

[Brockett, 1983] Brockett, R. (1983). Asymptotic stability and feedback stabiliaztion. *Differential Geometric Control Theory*, pages 181–191.

[Brooks, 1986] Brooks, R. (1986). A robust layered control system. *IEEE Journal of Robotics and Automation*, RA-2 No 1:14–23.

[Kolmanovsky and McClamroch, 1995] Kolmanovsky, I. and McClamroch, N. H. (1995). Developments in nonhlonomic control problems. *IEEE Control Systems*, pages 20–36.

[Martin, 1998] Martin, F. G. (1998). The handy board techinial referecne. *http://el.www.media.mit.edu/projects/handy-board.*

[Randal Beard, 1998] Randal Beard, Wynn Stirling, R. F. (1998). A hierarchical coordination scheme for satellite formation initialization. *AIAA Guidance, Navigation and Control Conference.*

[Stirling and Goodrich, 1999] Stirling, W. C. and Goodrich, M. (March 1999). Satisfycing games. *Information Sciences*, 144:255–280.