UMass MinuteBots 2017
Team Description Paper

Kyle Vedder, Edward Schneeweiss, Sadegh Rabiee, Samer Nashed,
Spencer Lane, Jarrett Holtz, Joydeep Biswas, David Balaban

University of Massachusetts Amherst, Amherst MA 01003, USA,
kvedder@umass.edu, eschnesweiss@umass.edu, srabiee@cs.umass.edu,
snashed@cs.umsass.edu, slane@cs.umass.edu, jaholtz@cs.umass.edu,
joydeepb@cs.umass.edu, dbalaban@cs.umass.edu

Abstract. This paper presents the electromechanical design, software architecture, and a selection of the algorithms used by the UMass MinuteBots, a first-year RoboCup team from the University of Massachusetts Amherst. The team uses RoboCup as a research application area for both hardware and algorithms. We are experimenting with more principled methods for evaluating and prototyping robots, and and are working on advancing the state-of-the art in multi-agent kino-dynamic navigation, time-optimal control, multi-modal state estimation, and role assignment.

1 Introduction

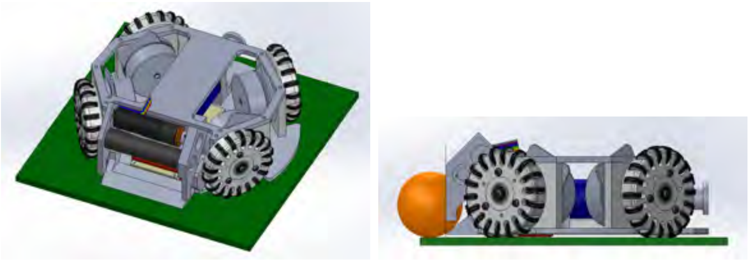
The UMass MinuteBots were founded at the University of Massachusetts Amherst in 2016. We are associated with the Autonomous Mobile Robotics Laboratory[1] in the College of Information and Computer Sciences. Our research focii in the domain of RoboCup SSL span several areas, including time-optimal control, joint kino-dynamic planning in adversarial domains, state estimation, and role assignment.

Our team's goal in the first year of participation at RoboCup is twofold: 1) design and build our first fleet of robots from scratch, and 2) develop robust control algorithms and develop the basic skills of robot soccer. As a new team, while much of our work involves reproducing existing state of the art, we nevertheless have a few novel contributions, including motion planning for true time-optimal control of omnidirectionl robots, and state estimation of system with multi-modal dynamics. This team description paper describes both our implementations of the state of the art, as well as our novel contributions, which will collectively prepare our team to compete at RoboCup 2017. In Section 2, we first present our hardware designs for the MinuteBots robots. Next, we present our STP [1] based multiagent coordination architecture in Section 3; followed by state estimation in Section 4; and motion planning and motion control in Section 5 and Section 6 respectively. Finally, we discuss our systematic software engineering in Section 7, and conclude in Section 8 with a roadmap of planned development till RoboCup.

[1] https://amrl.cs.umass.edu/

2 Hardware

Our software is designed ground-up to be agnostic of the platform hardware as far as possible. Hardware-specific translations, and model based compensation is deferred in our AI system to the hardware abstraction layer, which interfaces with the robots over radio, and ssl-vision. This allows us to interchangably run our system with our own custom designed robots, or on off-the-shelf SSL robots, the Yisibots, as well as simulated robots. The hardware abstraction has allowed us to develop and test our AI software on the Yisibots while concurrently working on the hardware design of our own custom robots. In this section, we describe the designs for our custom hardware, which is currently under prototyping assembly. The key design criteria for our robots is to optimize for simplicity of mechanisms, durability of components, and maintenance. Fig. 1 shows a rendering of the CAD model of the mechanical design of our robot, highlighting the low center of gravity of the design.



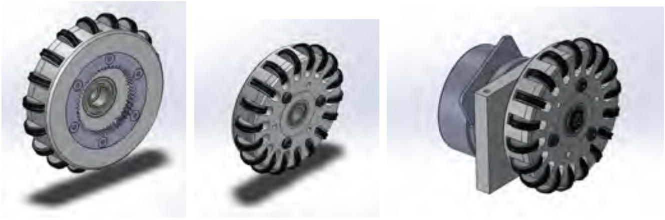
(a) Isometric view

(b) Side view

Fig. 1: A rendering of the mechanical design of the UMass MinuteBots.

2.1 Drive System

The drive system of the UMass MinuteBots robots consists of omnidirectionl wheels of diameter 55mm, driven by 50-Watt Maxon EC-45-Flat motors via a single stage 3.57 reduction gear. The front wheels are separated by 55° from the medial axis of the robot, and the rear wheels by 45°. With the chosen drive parameters, the robot has a theoretical maximum speed of 4.15m/s. The drive motors are equipped with integrated magnetic encodoers for feedback velocity control. In order to accommodate the width of the drive motors, the gear reduction consists of a pinion gear on the motor shaft, coupled with an internal gear that is integrated with the wheel hub. Fig. 2 shows a rendering of the CAD design of the wheel, and the drive sub-system.



(a) Wheel back (b) Wheel front (c) Drive assembly

Fig. 2: The wheel design and drive assembly

2.2 Kicking Mechanism

The design goals for the kicking mechanism are to optimize our system for momentum transfer from the kickers to the ball. The chip-kicker has a 45° front face that impacts the ball. Both the front face and the rear impact face of the chip kicker are concident with the hinge axis, thus minimizing shearing stress on the hinge pin. The main kicker consists of a flat kicking plate 45mm wide, and supported by a 90mm long rod.

2.3 Dribbling Mechanism

The dribbling mechanism is important for ball control, especially when receiving passes. Our design goals for the dribbler are to have an adjustable, durable mechanism, which is designed in a way that is easy to repair. Because the dribbler and kicking components experience the most fatigue, it is important to design these systems such that they are as durable as possible. When they do break, we would also like their design to be such that making repairs or manufacturing replacement parts is not unduly challenging.

2.4 Electronics

The electronics of the MinuteBots robots are designed to be modular, to aid in fabrication, debugging, and maintenance. There are three types of modules, which are mounted on a main control board: 1) four hall sensor brushless DC (BLDC) motor drivers with quadrature encoder based velocity feedback control for the four drive motors; 2) one sensorless BLDC motor driver for the dribbler motor; and 3) one kicker board to manage charging the kicker capacitors and actuating the kicking solenoids. The hall sensor BLDC motor drivers are based on the Allegro Micro A3938 BLDC motor controller, and the sensorless BLDC drivers are based on the Allegro Micro A4963 sensorless BLDC motor controller. Each sensored BLDC driver also includes an STM32042K6 microcontroller for on-board velocity PID control using quadrature encoder feedback. The main board consists of a single STM32042K6 microcontroller that 1) translates commands received over the radio to individual motor commands; 2) controls the charging, and kick triggering based on the commands received and the IR kicker interrupt; and 3) performs system monitoring and failure reporting, including motor health, kicker charge status, and battery status.

3 Multiagent Coordination

Our approach to multiagent coordination closely mirrors the Skills Tactics Plays (STP) architecture proposed by the CMDragons [1]. Skills represent low level behaviors such as motion control, dribbling, and kicking. These are the lowest level of the STP architecture and the last thing to be executed. We have developed skills for motion control, navigation deflection, kicking, and interception. We intend to supplement this with additional skills such as dribbling, turning while maintaining ball selection, and pass receiving. We focus on developing skills that are applicable to a wide variety of situations For example, the kicking skill is applied when a robot is passing and when a robot is shooting. The only difference between those two cases is the target that the ball is being kicked to.

The second level of the STP architecture is Tactics. Tactics are a higher level of abstraction than skills but are still executed for a single robot. We have implenented number of our tactics as finite state machines. The structure of one of these tactics, simple attacker, is shown in Figure 3. In addition to this transition structure, we define a set of preconditions and invariants for each state within the tactic.

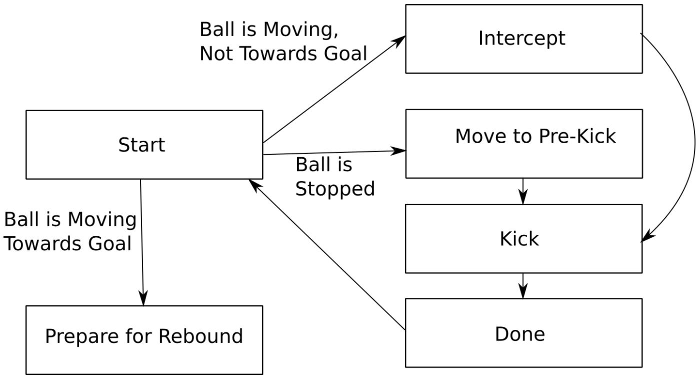


Fig. 3: Simple Attacker tactic structure.

Plays are the highest level within the STP architecture. Plays are defined as sets of roles and roles are defined as sequences of tactics. Like tactics, plays have sets of preconditions and invariants. When performing role assignment, a play is selected from the set of plays whose preconditions hold at any point and then each role is assigned to a particular robot. Each robot then executes the tactic that was assigned to it at that point. The flow of this overall process is shown in Figure 4.

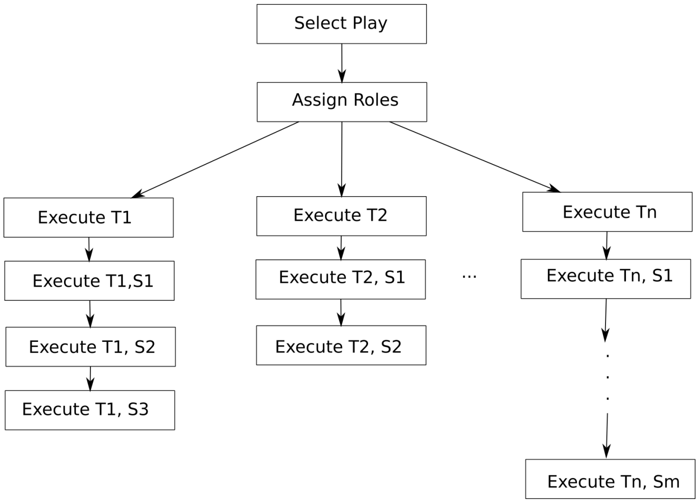


Fig. 4: Skills Tactics Plays structure.

4 State Estimation

State estimation is vital to the performance of a RoboCup soccer team. Robust and accurate state estimation allows for robust and accurate tactic execution. As an initial implementation we use an Extended Kalman Filter (EKF) to smooth the output of SSL Vision. This allows us to get a smoother estimate of the positions of each of the objects without much overhead.

We are currently experimenting with a hybrid model state estimation approach. We split the movement behavior of each of the objects into different modes and have separate state transition parameters for each of these modes. As an example, the movement of the ball is significantly different if it is being dribbled than when it is being chip kicked or flat kicked. An example of the different modes that the ball might be in is shown in Fig. 5. We define a set of transition regions between each of these modes. For example, the ball has likely transitioned from “stopped” to “flat kicked” when it is traveling in a straight line with a high velocity. We treat these discrete modes as hidden states and the SSL Vision messages as our observations. We intend to use a multi modal filter, such as the well-known Interacting Multiple Models (IMM) filter, to simultaneously estimate the hidden discrete and continuous state [2]. We believe that this will allow us to more accurately estimate the state of the ball and the robots when compared to simply applying the EKF.

5 Motion Planning

We have identified motion planning, control and safety to be three critical behaviors in the RoboCup domain. We believe that making these three behaviors

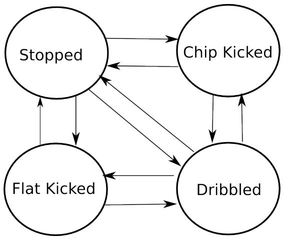


Fig. 5: Example of movement modes for the ball.

robust and accurate will make our team competitive at the actual event. We also have ongoing research in these areas.

For qualification and testing purposes, our initial approach to motion planning is to use a Probabilistic Roadmap (PRM) planner [3]. This allowed us to have reliable and fast motion planning with relatively little effort expended on implementation.

In the future intend to experiment with other approaches such as execution extended rapidly-exploring random trees (ERRT) [4] and a tangential planning approach similar to what STOx proposed [5]. In the months leading up to the competition, we intend to evaluate these planning approaches in a variety of conditions, select the one with the best performance, and make extensions to it in order to further improve the performance.

In order to supplement the navigation, help account for errors in state estimation, and improve safety guarantees during the course of a match, we also utilize Dynamic Safety Search (DSS) [6]. DSS is used as the last step of our planning/control loop. It is run after the controller has been calculated and is used to help prevent any crashes that might occur as part of the competition. This also helps minimize damage during our testing and development process.

6 Motion Control

Our initial approach to motion control is to use the Near Time Optimal Controller (NTOC) proposed by Kalmár-Nagy, DAndrea and Ganguly [7]. This controller allows for near optimal performance in problems which either start or end with the robot at rest. NTOC works by bounding the robot's acceleration and velocity magnitudes and solving 1D problems corresponding to each axis. Each 1D problem is given a portion of the maximum acceleration and velocity, determined by a binary search which requires both 1D problems take the same amount of time to complete.

We improve on this method with an algorithm we developed which can handle problems with arbitrary start and end velocities and with bounded acceleration and unbounded velocity. From the robot dynamics we derive a functional form of the time optimal acceleration with five parameters. Given the initial and final conditions of a problem, we use a non linear solver to search for the value of these parameters that cause the robot to reach the final conditions. We intend to use this control algorithm for cases which require quick action, such as ball intersection, and NTOC as a default control.

7 Software Engineering

Our design goal for our software architecture is to have a simple, robust and reliable system. We present the architecture of our design in Figure 6. Most notably, we are using an asynchronous architecture that allows us to perform state estimation, planning and control, and tactic execution on separate threads. Multithreading has the benefit of allowing our input processing to be resilient to slowdowns in messages from SSL Vision without negatively impacting other processing systems.

In addition, we have taken a disciplined approach to software development in order to ensure a high degree of software quality. We adhere to the Google C++ style guide in order to ensure uniformity, and we enforce this through use of a automatic compliance checker. We also treat all warnings as errors, as warnings are often indicative of bad practices and potential bugs. In order to gain a high degree of confidence in the correctness of our implementations, we heavily utilize test driven development. These tests are run on every build to highlight regressions the moment they occur in development.

We separate the radio server from main soccer program, with communication occurring over a multicast port using the standard SSL radio protocol messages. This facilitates the abstraction of our soccer program from the hardware layer.

To that end we have also have written a simulator to serve as a drop-in replacement for SSL Vision. Currently, we simulate position and velocity of every robot as well as basic ball mechanics, and use SSL Vision's graphical client to visualize the field. As mentioned above, this allows us to use the same soccer program on both simulated and real robots without any kind of code or configuration changes.

8 Roadmap and Conclusion

While we have many of the basic building blocks needed to be successful at the Robocup competition, we are a first year team and still have work to do in order to be truly competitive. On the hardware side, we have completed our design and are in the process of manufacturing our final hardware. We have a complete drive system prototype shown in Fig. 7, and are currently working on kicking mechanism prototypes. On the electronics front, we have finalized and prototyped our boards and are working on assembling a full board prototype. We are on schedule to complete the manufacturing of our robots in time for the competition in July.

On the software side, we have focused the last few months on developing the parallel architecture described above. We have developed a number of tactics and skills that are demonstrated in our qualification videos such as obstacle avoidance

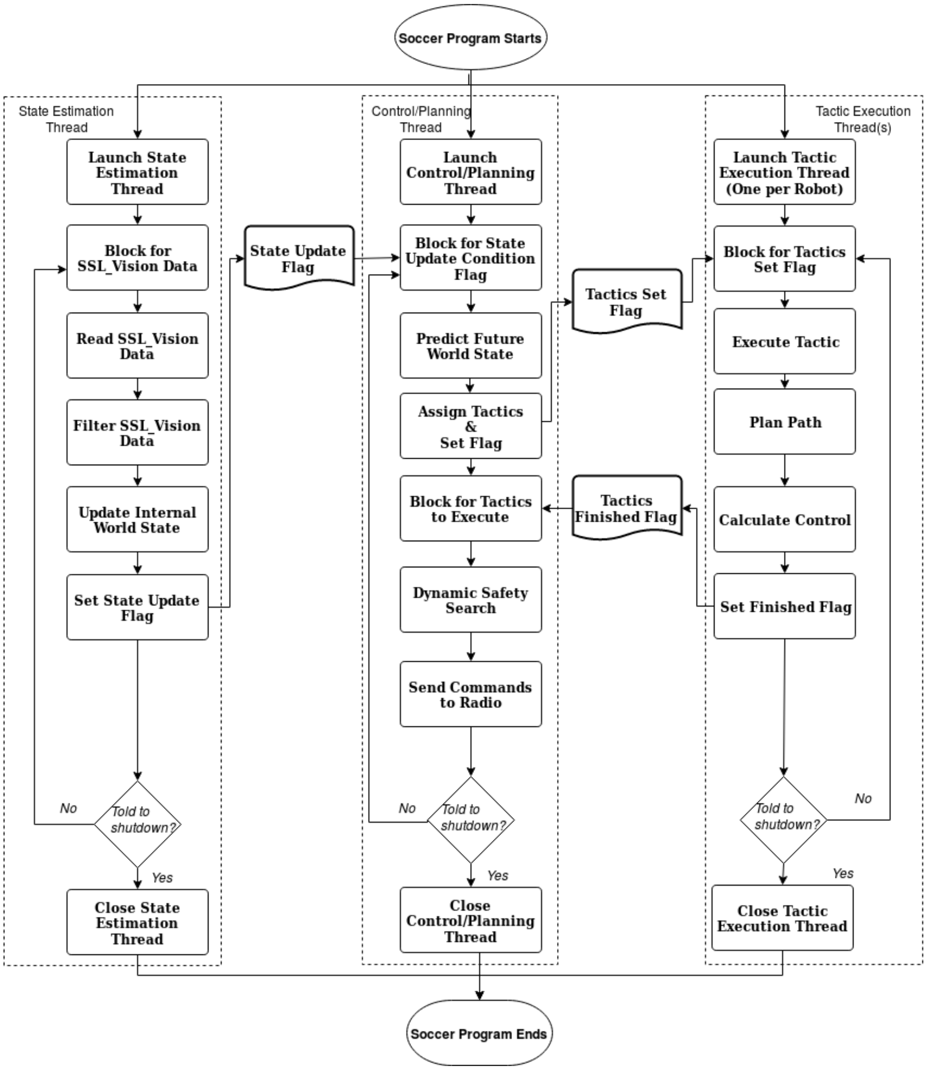

Soccer Program Starts
State Estimation Thread
Control/Planning Thread
Tactic Execution Thread(s)
Launch State Estimation Thread
Block for SSL_Vision Data
Read SSL_Vision Data
Filter SSL_Vision Data
Update Internal World State
Set State Update Flag
No
Told to shutdown?
Yes
Close State Estimation Thread
State Update Flag
Launch Control/Planning Thread
Block for State Update Condition Flag
Predict Future World State
Assign Tactics & Set Flag
Block for Tactics to Execute
Dynamic Safety Search
Send Commands to Radio
No
Told to shutdown?
Yes
Close Control/Planning Thread
Tactics Set Flag
Tactics Finished Flag
Launch Tactic Execution Thread (One per Robot)
Block for Tactics Set Flag
Execute Tactic
Plan Path
Calculate Control
Set Finished Flag
Told to shutdown?
No
Yes
Close Tactic Execution Thread
Soccer Program Ends

Fig. 6: Block Diagram for our Robocup Soccer Program

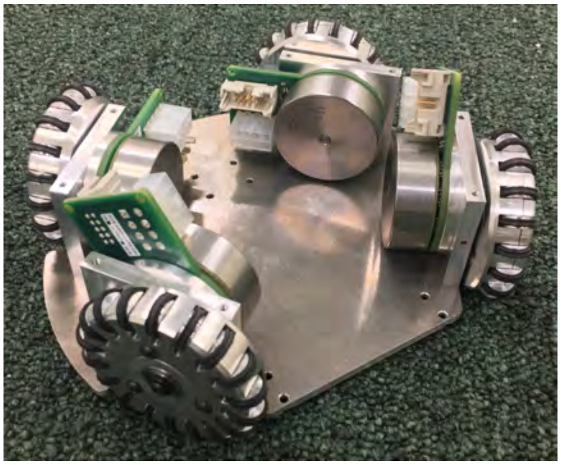


Fig. 7: Prototype of MinuteBots drive platform.

and shooting at open angles. In order to be successful at the competition, we will focus on developing additional skills and more robust tactics. In particular, we are going to focus on robust passing, threat based defense, and creative offensive strategies. Once we have completed robust implementations of these basic skills and tactics, we intend to focus our development on making our strategies reliable and robust enough to support competition.

References

1. B. Browning, J. Bruce, M. Bowling, and M. Veloso, "Stp: Skills, tactics, and plays for multi-robot control in adversarial environments," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 219, no. 1, pp. 33–52, 2005.
2. H. A. P. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients," *IEEE Transactions on Automatic Control*, vol. 33, pp. 780–783, Aug 1988.
3. L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, Aug 1996.
4. J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 2383–2388 vol.3, 2002.
5. S. Rodríguez, E. Rojas, K. Pérez, J. López, C. Quintero, and J. Calderón, "Fast path planning algorithm for the robocup small size league," in *Robot Soccer World Cup*, pp. 407–418, Springer, 2014.
6. J. R. Bruce and M. M. Veloso, "Safe multirobot navigation within dynamics constraints," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1398–1411, 2006.
7. T. Kalmár-Nagy, R. DAndrea, and P. Ganguly, "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robotics and Autonomous Systems*, vol. 46, no. 1, pp. 47–64, 2004.