

RoboPatriots: George Mason University 2011 RoboCup Team

Keith Sullivan, Christopher Vo, and Sean Luke

Department of Computer Science, George Mason University
4400 University Drive MSN 4A5, Fairfax, VA 22030 USA
{ksulliv2, cvo1, sean}@cs.gmu.edu

1 Introduction

The RoboPatriots are a team of three humanoid robots designed by the Computer Science Department at George Mason University. Each robot is based on the Kondo KHR-3HV, a customized Surveyor SVS camera, and a Gumstix embedded computer (see Figure 1(a)).

2 Hardware

We are interested in embodied AI, so we choose commercially available hardware rather than fabricating our own. After Robocup 2010, we realized our robots needed additional computational resources and an extra degree of freedom in each leg to allow rotation (see Figure 1(b)). Figure 2 shows the hardware architecture and information flow between components.

The robot base is the Kondo KHR-3HV. Each robot has 3 DOF per arm, 6 DOF per leg, and 2 DOF in the neck. The eighteen Kondo KRS-2555HV digital servos used in the arms and legs produce 14 kg-cm of torque at a speed of 0.14 sec / 60 degrees. The 2555HV servos communicate via a serial protocol and are controlled by the RCB-4 servo controller board. In addition, two KRG-3 single axis gyros and one RAS-2 dual axis accelerometer connect to the RCB-4. The two Kondo KRS-788HV digital servos used in our pan/tilt mount produce 10 kg-cm of torque at a speed of 0.14 sec. / 60 degrees. These servos are controlled by the Surveyor SVS vision system via PWM.

Our main sensor is the Surveyor Stereo Vision System (SVS) [1]. The SVS consists of two OmniVision OV 7725 camera modules connected to two independent 600 MHz Blackfin BF537 processors. The two camera modules are mounted on a pan/tilt mount with 10.5 cm separation. Each camera module operates at 640x480 resolution with a 90-degree field of view. The two processors are connected via a dedicated serial bus with hardware flow control.

For 2011, we added an additional computer: a Gumstix Overo Air [2]. The Air is a 600 MHz OMAP 3503 processor with 256 MB of flash and 256 MB RAM. The Air runs embedded Linux, and provides 802.11 b/g. The Air communicates with the RCB-4 over a dedicated serial bus with a custom inverter circuit for logic level shifting and signal inversion. The SVS and Air are connected via

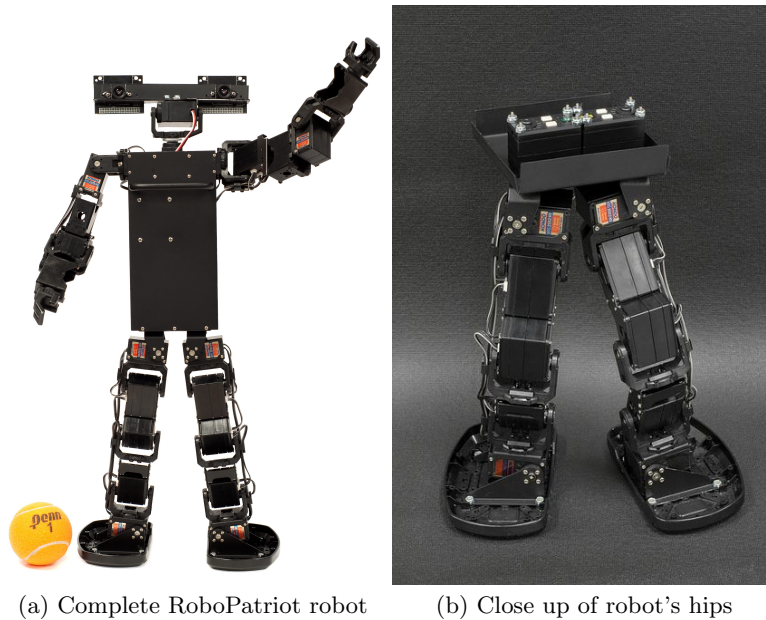


Fig. 1. 2010 RoboPatriot's robot and close up of new rotation servos in the hips.

an SPI bus. The Air and SVS are mounted on a custom motherboard which also provides power distribution, USB connections, and sensor connections. See Figure 3 for a prototype.

Each robot has a 11.1V 2200 mAh battery.

3 Software

The goalie and attackers each run a state machine for high level decision making. The state machines include states for approaching the ball, orientation for kicking, and kicking towards the goal. As in past years, predefined motions are stored on the RCB-4 and are not dynamically modifiable. However, we can interrupt motions during execution and can run cyclic motions for an arbitrary length (e.g., we can execute N walking steps based on dynamic sensor information). The software architecture is split across the RCB-4, SVS and Gumstix as follows:

- The RCB-4 Servo controller handles gyro stabilization and execution of predefined motions.
- The SVS performs vision related tasks (discussed below) and localization.
- The Gumstix detects falls, handles communication, and runs the state machine.

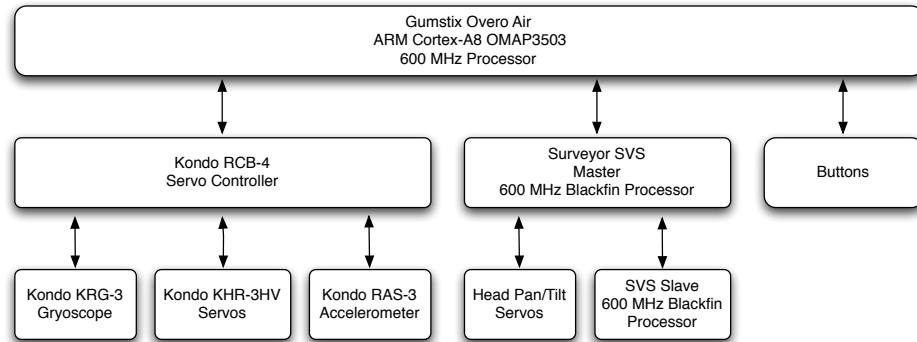


Fig. 2. The hardware architecture of the RoboPatriots, and the information flow between components.

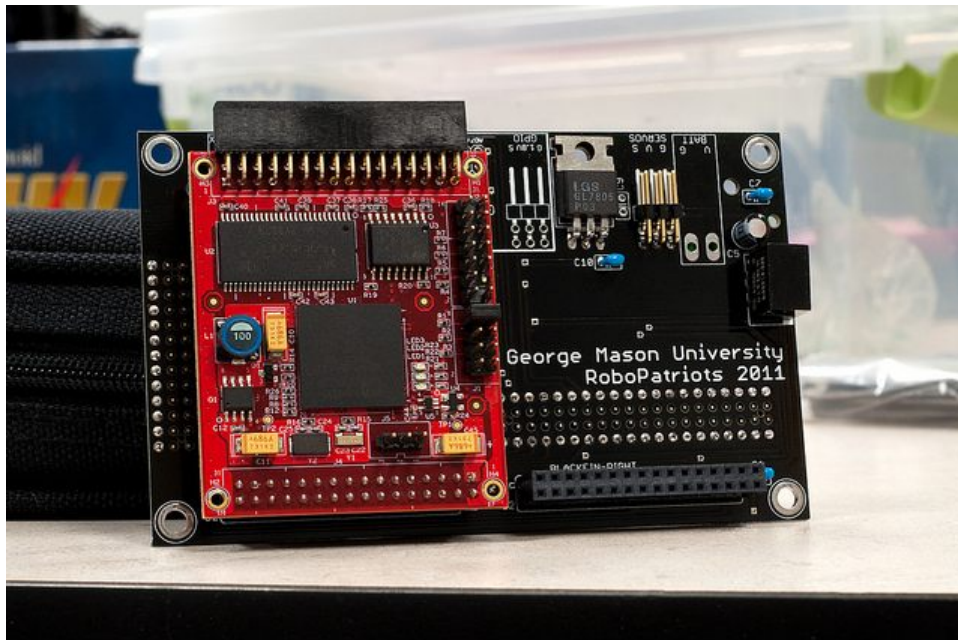


Fig. 3. A prototype of the integrated motherboard connecting the Gumstix computer and SVS modules.

Vision One camera module of the SVS handles basic color tracking, first using its own image, then the image from the other camera module. If both camera modules detect the color, then stereo depth mapping combined with the camera pan/tilt position provides an approximate physical distance to the object of interest. In addition, ground-plane calibration allows us to ensure detected objects are on the floor, and shape detection ensures detection of appropriate objects such as the goals, the ball, and field lines.

Localization Our localization module uses a particle filter with random particle injection. The field is represented as a topological graph, where nodes are distinct field features (line intersections, goal posts, etc.), and edges are logical connections between features. The sensor model computes the probability of observing a set of features given a location. The motion model is based on basic walking behaviors. The localization information is used for autonomous repositioning and robot coordination.

4 Learning from Demonstration

Due to the difficulty in developing robust behaviors for RoboCup, we recently started experimenting with *learning from demonstration* as a way to train robots to play soccer. In learning from demonstration, a robot learns a behavior based on real-time samples from a demonstrator. Learning from demonstration is an attractive alternative to the traditional coding, trial, and debugging cycle. In our approach, the robot learns the transition between states in a hierarchical finite state automaton where transitions are based on sensor information.

One challenge facing learning from demonstration is the number of samples required to learn robust behaviors. In a simulated environment, gathering sufficient data is not particularly difficult. However, collecting samples using robots is quite expensive: each sample is a data point from an experiment conducted in real-time. We recently developed a learning from demonstration system capable of rapidly training virtual agents to perform complex behaviors with a minimum number of samples [3].

Our system learns behaviors in the form of hierarchical finite state automata (HFAs): the individual states are either pre-coded basic behaviors (“go forward”, “turn left”, etc.), or learned HFAs, and the each transition function is learned by a classification algorithm on *features* gleaned from robot sensors, internal state, etc. Each feature can take any form allowed by the classifier. Once learned, the HFA is then saved and then can be used as a state in a more complex behavior. This approach is supervised and allows robots to learn both stateless and stateful behaviors.

Our ultimate goal is to field a humanoid soccer team which is trained, rather than programmed. To that end, we have ported our learning from demonstration system to the RoboPatriot humanoids and have trained the robot to perform visual servoing [5]. The goal was for the robot to search for the ball by turning the “correct” direction, and walk towards the ball. Using feature information from

the camera, a group of computer science graduate students with no humanoid robot experience successfully trained the robot. In addition, we demonstrated that learning complex behaviors in a hierarchical fashion is quicker and easier than learning complex behaviors in a monolithic fashion.

Transitioning from a single robot to a group of robots, we have also organized a team of robots into a *robot hierarchy*, with robots at leaf nodes and *coordinator* robots as nonleaf nodes [4]. This tree-structured organization dovetails with our HFA-based task decomposition. Individual robots are trained as usual, with the caveat that all robots share the same behavior library. Coordinator agents control a group of agents, and themselves are trained to develop an HFA. Future work will focus on heterogeneous robot hierarchies: each subgroup runs a different HFA, with dynamic subgroup membership.

References

1. Surveyor stereo vision system. http://www.surveyor.com/stereo/stereo_info.html (2010)
2. Gumstix inc. <http://www.gumstix.com> (2011)
3. Luke, S., Ziparo, V.: Learn to behave! rapid training of behavior automata. In: Grześ, M., Taylor, M. (eds.) Proceedings of Adaptive and Learning Agents Workshop at AAM AS 2010. pp. 61 – 68 (2010)
4. Sullivan, K., Luke, S.: Hierarchical multi-robot learning from demonstration. In: Proceedings of Robot Science Systems Conference (submitted)
5. Sullivan, K., Luke, S., Ziparo, V.A.: Hierarchical learning from demonstration on humanoid robots. In: Proceedings of Humanoid Robots Learning from Human Interaction Workshop. Nashville, TN (2010)