

Satrap Team Description Paper

Behnam Azimi, Behzad Azimi, Mohammad Kashani, Shahin Mohammadi, Hossein Mohammadkhan, Ali Tayyebi, Amin Yazdani, and Pedram Zohouri

Satrap Group
Tehran, Iran

{Azimi, Behzad, Kashani, Mohammadi, Mohammadkhan, Tayyebi, Yazdani,
Zohoori}@Satrapteam.ir
<http://www.Satrapteam.ir>

Abstract. This paper is aimed to summarize the total process which is done in order to design the Satrap robots, including hardware and software specifications. This is the third year that Satrap team participating in Robocup MSL competitions. This year researches focused on image understanding, and improve world modeling of the agents. Results submitted to Robocup symposium 2006 for qualification.

1 Introduction

Robocup soccer competitions are mainly held to develop a test bed for implementing novel ideas in AI field. Competitions has immense results during last few years, but all of these achievements are leading separately. This have a sensible side effect, which is making a gap between simulation league and other leagues. In designing real robots, electrical and mechanical problems, and current mechanical limits from the other hands, usually avoid participants to focus on designing autonomous agents. It seems an important need to merge these researches, and as a point to start we need to model the outside world considering its noise. This year we focused on creating world model from sensors, and we are currently working on methods to inspiring the agent knowledge-base on its perception. We first take a look at overall hardware specifications in this article then we show a quick overview of the process needed to extract features from visual sensors to update the world model, and decision making according the world state.

2 Hardware

In this part, we are going to discuss about physical abilities of Satrap robots, as currently designed. Our agent hardware specifications are as follows, expecting that the ball handling and kick method may be changes significantly, but changes are not complete yet.

2.1 Drive

Our robots are using 3 active wheels without any passive wheel, which give them the ability to directly drive in 6 directions, and moving on an arbitrary direction with at most 30 degrees rotation. Current design of agents, is the extension of our old 4-wheeled vehicle which was used in Robocup2004- Lisbon[1].



Fig. 1. Abstract Driving System

Maximum speed in a direction can simply be calculated by projecting speed vectors in that direction and summing them. The result shows that the maximum velocity of the agent can reach approximately 4 (m/s), with operating two of the motors with max. vel. in the same direction. The motors' power is about 135 (Watt) and they need a 24 (V) power supply to drive.

2.2 Motor controller

Motor controllers use a PIC16F767A microcontroller chip to handle the locomotion of the agents. Each of the robots can change the duty cycle of their PWM pulse to vary their velocities, which means the agent has a fuzzy motion. The controller board connects to the notebook system using a serial communication. The Motor Handler Module is a high-level program executed on PCs, which is responsible for interconnecting low-level motor controllers and high-level actions needed for the path planning module.

2.3 Kicker

Current agent kicker uses pneumatic system with an air capsule, which can approximately handle 25 kicks for each agent. Each agent has a passive ball handler, a very simple but still useful method [2]. We are integrating an elastic rope with our ball handling system, in order to graduate the momentum transfer between ball and the agents.

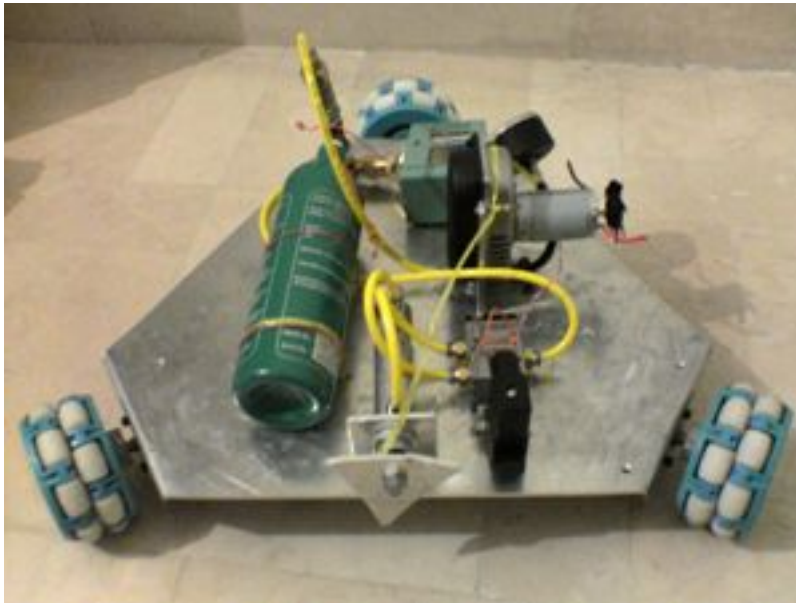


Fig. 2. Abstract Kicker System

2.4 Power Supply

We are using 14 batteries per agent, each of them 3.7 (Volt), 1200 (MilliAmpere/Hours), Li-Ion. Two of them are use for microcontroler and opto-coupler boards. Other twelve batteries are join together to make two basic power levels in our circuits, 12 (V) and 24 (V). The 12 (V) used for launching motor driver's board, while the 24 (V) used for power supplying of the motors.

2.5 Sensors

Agents are using ultra-sonic range finder sensors to find out distance of other objects in the field. They also use a compass to estimate their orientation, which used as a feedback to eliminate motion noises in motor controllers. Same as most

other teams, agent uses an omni-directional vision sensor[3][4], to grab images in 360 degrees around it. Their camera attached to notebook using firewire, and can grab images in a rate about 30 (f/p) in RGB format.



Fig. 3. Omni-Vision Snapshot

3 Software

This year the total software framework is written in C Sharp. DirectShow is used as its framegrabber. Each agent have an instance of agent class, which initially connects to the monitor program in construction time. Monitor broadcasts the referee box commands to the agents, also monitor the positions of them. It also implements emergence Stop/Start button to remotely stop the robots in emergence cases. Further comes more details about different blocks of software in separate sections.

3.1 Image Understanding

This is one of the most important modules in our programs. It aims to extract objects from the sequence frames grabbed from camera, and estimating objects position. It is a vital need for decision making to know this information considering noises. Image understanding is a two way interconnection between world model and Image Processing Module[5]. IPM analysis images, extract objects and update world model, while the state of the world model could help IPM to reduce its computations and prediction of objects in next steps. There are three useful consequence comes from this approach:

- It used to predict the state of world after n-steps according to the current state. In simulation league, it is usually know as opponent modeling.

- It used to define ROIs which objects may be found, instead of processing the whole image, we do an initial search for initializing the ROIs, Then just "Track" the objects.
- It used for noise reduction. As the motions are always smooth, we expect a predictable behaviour of the object. So unacceptable moves could simply recognized and used it for enhancement of perception.

After extracting objects, we need to find out their distance in the real world. Our approach to this problem is Supervised Learning using ANN[6][7]. It uses some sample images and their real distance as samples for its network, and use a back propagation for balancing the weights to interpolate the distance function.

3.2 World Modeling and Decision Making

After representing the objects, we need to update the world model. In updating the world model we have to stick some fact in mind, that's we can predict the next step from current state. As mentioned in previous section, it's a very important ability in avoiding the noises and also in decision making. For example when we are going to write a strategy manager, we can predict the fastest player which can catch the ball. Other players could stay in their defined positions. Strategy manager uses an expert coach[8] to predict the opponent behaviour and offer an offensive/deffensive strategy.

3.3 Path Planner

Path planner works in conjunction with decision making module, to offer a safe path from the current state to the opponent goal. It works simply by assuming that the agents must:

- Avoid collide with obstacles in the field, including team mates and opponents
- Agents must always trace the shortest path to the goal

To reach above goals, we have two steps. First we need an estimation of the robot positions. Next we assume a line directly from the current position to the opponent goal, then we have to increase the order of interpolating polynomial by reaching near to each obstacle. We can use Spline interpolation algorithm[9] to define these polynomials. It is obvious that opponent modeling is not a deterministic method, so we may need some online corrections according to opponent actions.

4 Conclusion

In this paper we showed an overview of the Satrap robots design. Most of the efforts focused on developing modular autonomous agents, which can simply extend. We are aimed to write a base platform for implementing MSL robots in next year, which could be simply be used by new comers.

References

1. Azimi, B., Barahmand, Z.: Omni- directional wheels control analysis for 4-wheeled robots locomotion. The First National Mechatronic Conference- Iran (2002)
2. Azimi, B., Barahmand, Z.: Track wheeled ball handling system (twbhs) for middle-sized soccer robots. 3rd Mechatronic conference, Iran (2004)
3. Mohammadkhan, H.: Omni-vision for middle size robots. 1st Machine Vision conference, Iran (2003)
4. Winters, N., Santos-Victor, J.: Omni-directional visual navigation (1999)
5. Mohammadi, S.: World modeling using imaging sensors in middle-size league. Submitted for Robocup Symposioum (2006)
6. Mohammadi, S., Yazdani, A.: Supervised image processing using artifitl neural networks. 2nd Machine Vision conference, Iran (2004)
7. Lin, C.T., Lee, C.S.G.: Neural-network-based fuzzy logic control and decision system. *IEEE Trans. Comput.* **40**(12) (1991) 1320–1336
8. Carmel, D., Markovitch, S.: Opponent modeling in multi-agent systems. In Weiß, G., Sen, S., eds.: *Adaptation and Learning in Multi-Agent Systems*. Springer-Verlag: Heidelberg, Germany (1996) 40–52
9. Antia, H.M. In: *Numerical Methods for Scientists and Engineers*. (2002) p150