Satrap Team Description Paper

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

Satrap Group
Tehran, Iran
{Azimi, Behzad, Hazar, Kashani, Mohammadi, Mohammadkhan, Tayyebi, Yazdani, Zohoori }@Satrapteam.ir
http://www.Satrapteam.ir

Abstract. This paper is aimed to summarize the overall process which is done by Satrap team in order to prepare for participating in Robocup MSL competitions 2007, including hardware and software specifications. For the fourth year competing, this year researches are focused on optimizing both hardware and software systems. We tried to develop a completely independent localizer kit based on SLAM algorithms. Our image processing and world modelling modules are experimentally improved. The design of ball kicker and ball handling systems are changed, in order to improve kicking accuracy and handling reliability.

1 Introduction

Robocup soccer competitions are mainly held to develop a test bed for implementing novel ideas in AI fields. Competitions have immense results during last few years, but all of these achievements are leading separately in simulation leagues and robot leagues which make them completely apart. In designing real robots, electrical and mechanical problems and on the other hand current mechanical limits, usually avoid participants to focus on designing autonomous agents. It seems that the competitions need to combine these researches. To reach this goal this year we focused on modelling the MSL environment using data taken from our sensors and designing knowledge-base strategies based on agents perceived world models. In this article we first take a look at overall hardware specifications and then we show a quick overview of the process needed to extract features from visual sensors to update world model, and finally decision making according to world state.

2 Hardware

In this part, we are going to discuss about physical abilities of our robots, as currently designed. Agents hardware specifications are as follows, expecting that the ball handling and kicking 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 an improved design of our old 3-wheeled vehicle which was used in Robocup2005- Osaka.

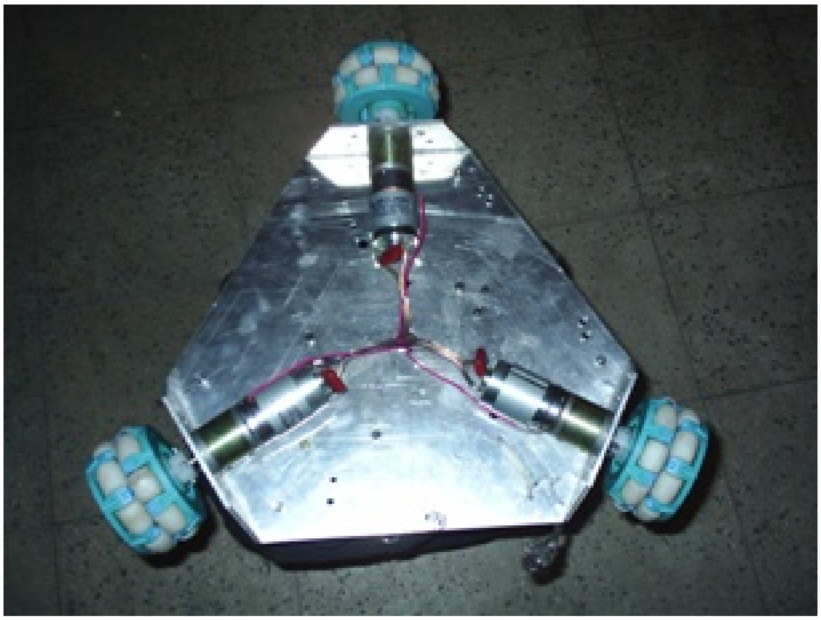

Fig. 1. Abstract Driving System

Maximum speed in a straight path can simply calculated by projecting speed vectors in that direction and summation of them. Result shows that maximum velocity of the vehicle can reach approximately 4.5 (m/s), with operating two of the motors with maximum velocity in the same direction. Motors power are about 135 (Watt) and they need a 24 (V) power supply to drive.

2.2 Motor controller

Motor Controllers uses a PIC16F877A microcontroller chip to handle the locomotion of the robots. Each of the robots can change the duty cycle of their PWM pulse to vary their velocities, which means it has a fuzzy motion. Controller board connects to the central processing system using a ethernet connection. Motor Handler Module, is a high-level program, which is responsible for interconnecting low-level motor controller, and high-level actions needed for path planning module.

2.3 Kicker

Current agent kicker uses strengthened celluloid system with an air capsule, which can handle up to 100 kicks for each agent. Each agent has a passive flexible ball handler, which designed to dynamically control the ball even in fast moving and swathing. We are integrating an elastic rope with our ball handling system, in order to graduate the momentum transfer between ball and the agents.

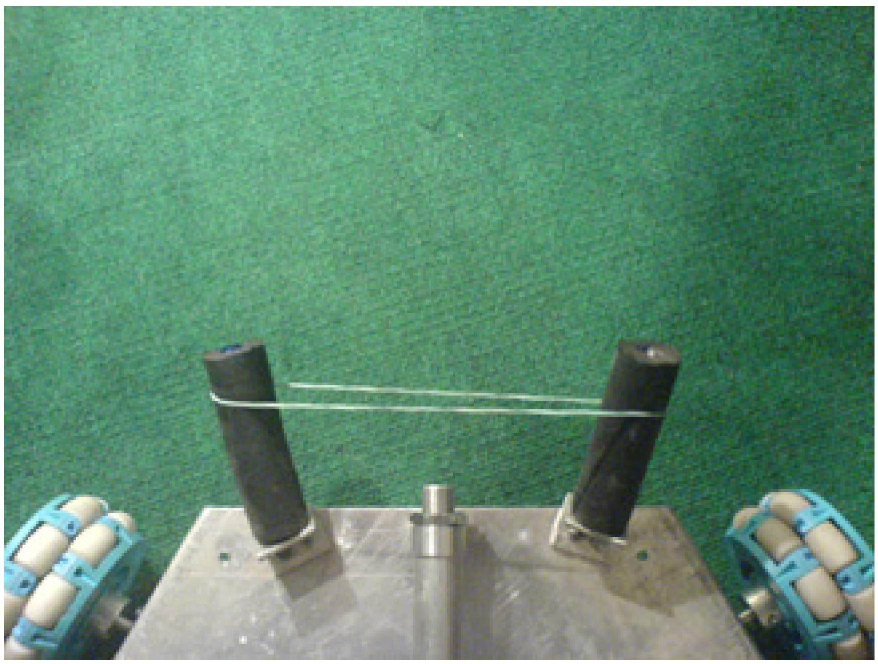

Fig. 2. Abstract Kicker System

2.4 Power Supply

We are using 5 batteries per agent, 3 of them are 12 (Volt), 7000 (MilliAmpere/Hours), sealed-acid and the others two batteries are Li-Ion 3.7 (Volt), 1200 (MilliAmpere/Hours) which are used for microcontroller and opto-coupler boards. The sealed-acid batteries make two basic power levels in our circuits, 12 (V) and 24 (V). The first is used for launching motor driver's board and the other 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[1][2], to grab images in 360 degrees around it. Their camera attached to the central processing unit using firewire, and can grab images in a rate about 30 (fps) in RGB format.

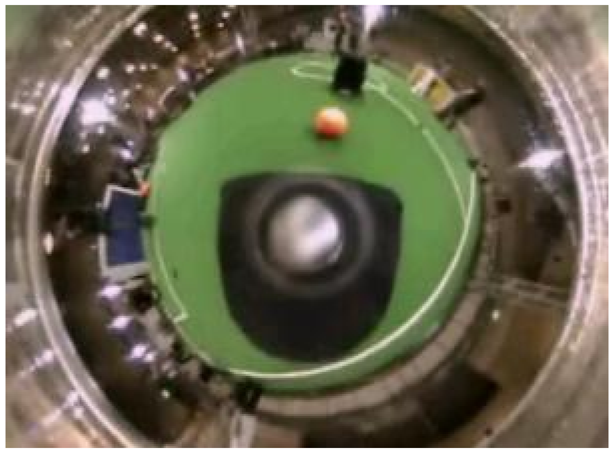

Fig. 3. Omni-Vision Snapshot

3 Software

This year the total software framework is linux-based and written in C++. Coriander is used to capture raw data from IEEE 1394 camera and redirect it to V4L loopback adapter. Localization module use these acquired frames to find objects position including obstacles positions and the agent itself. Path planning algorithm uses the topological map of the field to find a collision free path to opponent goal. Each agent tries connecting to the monitor station in its init step. Another thread is responsible for listening to referee box commands and affects them in decision making system. Further comes more details about different blocks of software in separate sections.

3.1 Image Processing

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's s a vital need for decision making system to update information considering noises. Image understanding is a two way interconnection between world model and Image Processing Module. 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 modelling.
- 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 behavior 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 Simultaneous localization and mapping (SLAM)[3]. It uses imaging and ultra sonic sensors to make a topological map of the environment.

3.2 World Modelling And Decision Making

After representing the objects, agent needs to predict the positions of objects. Noise removal is then achieved by comparing predicted positions and observed positions. Updating the world model is done with corrected positions. Decision making system is used world model data to find the best player for catching ball based on directions and distances of agents from the ball. Other agents positions are defined by our strategy manager module which uses an expert coach[4] to predict the opponent behavior and offer an offensive/defensive 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 the above goals, we build up a graph based on the obstacle agents positions and the possible routes to the goal through them, and find the shortest path by using Dijkstra shortest path algorithm. After reaching every node in the shortest path the graph is updated dynamically according to changes in the world models and the path will be rebuilt, if it's needed.

4 Conclusion

In this paper we have presented an overview of our robots design. Most of the efforts have been focused on developing modular and autonomous agents, which could be simply extended. We are aimed to build a base platform for implementing MSL robots in next year, which could simply be used by newcomers in this field.

References

1. Mohammadkhan, H.: Omni-vision for middle size robots. 1st Machine Vision conference, Iran (2003)
2. Winters, N., Santos-Victor, J.: Omni-directional visual navigation (1999)
3. Wang, C., Thorpe, C.: Simultaneous localization and mapping with detection and tracking of moving objects (2002)
4. 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