AT-manoid Team Description Paper 2011

Hamid Maleki\textsuperscript{1}, Alireza Amini Sohi\textsuperscript{1}, Mohammad mazinani\textsuperscript{1}, MehdTorshani\textsuperscript{1}, Elmira Imanloo\textsuperscript{1}, Aysan maleki, Afshin Aryayi nejad\textsuperscript{1}.

Amini.alireza1@gmail.com
WWW.RoBolad-co.ir

Abstract. The AT-manoid kid size humanoid soccer team is described in this paper. AT-manoid intends to develop an intelligent team of humanoid robots, which are able to play soccer. The project is described in two main parts: robot Hardware and robot Software. At first, mechanical structure of the 583 mm tall, autonomous humanoid robot with 21 degrees of freedom is described. Then, electronic architecture and electronic components such as processor, sensors and actuators are presented, and finally the software design including image processing system, walking and stability and ball decision making will be described.

Keywords: Humanoid Robot, MLP, ZMP, Artificial Neural Network

Resume

The goal of RoboCup organizers and competitors is for “a team of robots to be able to defeat a world champion team of human soccer players by the year 2050”. In order to achieve this goal and attraction of this scientific field for researchers, studying and research for building a humanoid robot have been started in AT-manoid technical and vocational training School since 2008. AT-manoid team is developed by a group of faculty members and students of Almme Tabarsi University, Iran. Team members have done research on industrial Rescue robots and specially artificial intelligence of these robots, 2D and 3D soccer simulation and deminer robots in recent years. Several team members have already contributed in the past RoboCup competition. (Awarding 5th place in 2D league of Iran open 2007, 1st place in 2D league of Brazil open 2008, 2nd place in 2D league of Iran Khwarizmi 2009, 6th place in Rescue real league of Iran open 2010, 1st place in Rescue league of Iran Khwarizmi 2010, 4th in 2D league place Iranopen 2011 According to skills and experience of team members .we have started building Our first kid-size two-legged humanoid robot, AT-manoid, since 2010.

1 instruction

AT-manoid robot has significant and powerful computational capabilities include image processing system, walking and stability and ball decision making.

1.1 Image Processing System

Real time image processing plays an important role in overall efficiency of humanoid soccer robots. Faster process of captured image from the camera includes finding of ball, lines, cross-lines and goals in the game-fields. In this section architecture of the developed image processing system is described. The vision system is divided in two parts: Low level and High level. In low level part, output of the camera module which contains a JPEG stream is transferred to the main processor via RS232. Because the RGB color space is not reliable in different lighting
conditions, captured images are transferred from RGB to YUV color space. The Second step is image segmentation. Here we have developed a segmentation algorithm for color images based on a multi layer neural network (MLP) that is trained with back propagation algorithm (BP). In this step the main feature of the image are fed to the network for training in offline mode. Then we create a look-up table (LUT) based on the result of neural network.

Training the images and extracting main features of the image both are done in offline mode that considerably decrease the process time in on-line mode. Also results and experiments have been shown that proposed algorithm for segmentation outperforms the traditional thresholding algorithm. In the high level stage, binary image analysis is used for blob formation. Using blobs important information such as ball and goal position and lines can be determined for each robot. The implemented vision system is capable of processing 320*240 images with the speed of 40fps.

1.2 Walking and Stability

One of the main challenges in implementation of the humanoid robot is finding a suitable walking-pattern model, so that the robot will be able to execute and run the issued command without falling down. Several walking-pattern methods have been studied, including Zero Moment Point (ZMP) walking, rhythmic walking, ballistic walking.

The first step in the method that has been used in our robot is implementing a walking pattern based on ZMP. In ZMP method, the aim is finding the point, where the center of gravity is maintained over the continues movement of robot in all times [1,2,4]. This method can be developed by mathematic model of kinematics of each part that have a degree of freedom. In this model for each degree of freedom a 3-dimension rotation matrix is considered and for each joint a translation vector can be used. Using the mentioned method can impose a huge calculations load on CPU for each step. We can decrease this calculation load by using a learning procedure. An artificial neural network is utilized for realizing this idea. After training the neural network with the extracted parameters from ZMP method, the robot will be able to calculate the next step parameters without using the ZMP method.

It can be shown that the problem of calculating the next step parameters using ZMP method is a structured problem and can be learned to a machine by a learning process. Fig.3 shows the method of extracting the ZMP parameters. After initial modeling of the robot different movement and walking patterns that is needed for robot in playing environment, is applied to it. Then the next step parameters for each movement and walking patterns is calculated using ZMP and stored in a database.

The next step is training a neural network with the available inputs (movement command) and outputs (ZMP parameters) in database. After applying the new correction on error the system is trained again. This procedure is repeated until we reach to a desirable error value.
By developing a trained neural network now, it’s possible to apply the movement command directly to the trained network. As it can be seen in figure 4, when a movement command is issued, first this command is analyzed by the trained network and then the required movement parameters for the next step are extracted and given to the robot.

Considering the fact that the problem is an interpolation of current position and the next position in one step a RBF (Radial Basis Function) is developed for the system as the neural network [3].

Fig. 4. Using Trained network for direct estimating of the ZMP parameters

1.3 Ball Decision Making

One of the main challenges in implementation of the humanoid robot in soccer game is ball decision making. In AT-manoid ball decision making is being developed based on Learning Classifier Systems. Figure 5 shows AT-manoid Kicking the ball.

Learning classifier systems learn to solve a problem by interacting through an online adaptation process that is very much similar to the trial-and-error interaction. At discrete time steps, a classifier system perceives the current state of the problem through its detectors and based on its current knowledge (on the classifiers in the population) it selects an action to perform on the problem through its effectors. Depending on the effect that the performed action has on the problem, the system receives a numerical feedback which can be either interpreted as the fuel of a motivational reservoir or as the incoming reward signal [5, 6, and 7].

Many learning classifier system (LCS) implementations are restricted to the binary problem realm (XCS). Recently, the XCS classifier system was enhanced to be able to handle real-valued inputs among others. In the real-valued enhancement, XCSF applies as a function approximation system that partitions the input space in subspaces specified in the classifiers [8]. Everything in a learning classifier system revolves around the population of classifiers which identifies what the system knows about the problem solution. Each classifier represents a nugget of such knowledge. In XCS and XCSF, a classifier consists of (i) a condition, which identifies an area of the problem domain; (ii) an action, which represents a decision on the sub problem identified by its condition; (iii) the prediction, which estimates the return that the system expects when the classifier is used; (iv) the error, which estimates the average error of the prediction; and (iv) the fitness, which estimates the relative accuracy of the prediction and it is a function of the error [7]. The simplified architecture of a learning classifier system is shown in Figure 6. At each time step, the classifier

Fig. 6. Simplified architecture of a learning classifier system
system perceives the current state of the problem through the simplified architecture of a learning classifier system is shown in Figure 6. At each time step, the classifier system perceives the current state of the problem through its detectors; it builds a match set containing all the classifiers in the population whose condition matches the current sensory input. The match set typically contains classifiers which advocate contrasting actions; accordingly, the classifier system evaluates each action in the match set and selects an action to be performed balancing exploration and exploitation. The classifiers that advocate the selected action are put in the action set and the selected action is sent to the effectors to be executed [7].

Depending on the effect that the action has in terms of problem solution, the system receives a scalar reward. The incoming reward is distributed to the classifiers that are accountable for it through the credit assignment component. In the early classifier system models, this step was implemented by Holland’s bucket brigade algorithm, a kind of temporal difference learning [9]. In XCS and XCSF [5, 6], credit assignment is implemented by a modification of Q-learning. On regular basis, the rule discovery mechanism is activated with the goal of improving the current solution, which is finding better classifiers, by recombining and mutating the classifiers in the population. For this purpose, a genetic algorithm is applied to the classifiers in the action set. Two classifiers are selected, recombined, and mutated. The resulting offspring classifiers are inserted in the population while other two are deleted to keep the number of classifiers in the population constant [7].

Applying new genetic programming approach for classifier population mutation and improve XCSF credit assignment mechanism are the related works to classifier systems that we are considered.

References


Youtube Video:  
http://www.youtube.com/watch?v=lICfIxgwxs