Kouretes 2008 Team Description Paper
Nao MSRS Simulation Competition

Team Kouretes - MSRS Simulation Competition

Technical University of Crete (TUC), Chania 73100, Crete, Greece

Abstract. Kouretes has participated in the MSRS (Microsoft Robotics Studio) Simulation Competition at Robocup 2007 and this year at Robocup 2008. This short paper gives an overview of the team and describes ongoing work.

1 Team History

Team Kouretes was founded in February 2006 by Michail G. Lagoudakis and became active in the Four-Legged league. In January 2007, under the leadership of Nikos Vlassis team activities were extended to the Simulation league. The team had its first exposure to RoboCup at the RoboCup 2006 event in Bremen, Germany, where it participated in the Technical Challenges of the Four-Legged league. At that time, Aibo programming by the team was done exclusively in an interpreted language, the Universal Real-Time Behavior Interface (URBI), without any use of existing code. Subsequent work led to the participation of the team in the Four-Legged league of the RoboCup German Open 2007 competition in Hannover, Germany. The software architecture of the team was developed on the basis of previously released code by GT2004 and SPQRL 2006. The tournament included ten teams from all over the world. Kouretes reached the quarterfinals round, where it was defeated by the 2006 World Champion Nubots. The team ranked in the $7^{th}/8^{th}$ place in a tournament featuring the team's first win and first goals. In Spring 2007, the team began working with the newly-released Microsoft Robotics Studio (MSRS). The team's software was developed from scratch exclusively in C# and included all the required services, as well as the motion configuration files for the simulated RobuDog robot of RoboSoft. The team's participation in the MSRS Simulation Challenge at RoboCup 2007 in Atlanta led to the placement of the team at the 2^{nd} place worldwide bringing the first trophy home. The tournament involved nine teams from all over the world; Kouretes was the only European participating team. In October 2007, Team Kouretes and Team Cerberus (Turkey) were invited to play friendly demonstration games for the public during the international business meeting Hi-Tech Innovators Partenariat 2007 in Thessaloniki, Greece. This two-day event marked the first time full RoboCup games under the official rules were played in Greece. In May 2008, Team Kouretes was invited to Rome, Italy by Team SPQR to play friendly demonstration games for the public at the TechnoTown Museum and to participate in RomeCup 2008. Detailed information about the team, including pictures and videos, may be found at the team's site `www.kouretes.gr`.

2 Team Leadership

Michail G. Lagoudakis is an assistant professor with the Division of Computer Science of the Department of Electronic and Computer Engineering (ECE) at the Technical University of Crete since 2005. He received his Ph.D. degree from Duke University, USA in 2003 and was a postdoctoral researcher at the Georgia Institute of Technology, USA until 2005. His research experience in robotics spans several areas: path planning, motion control, reinforcement learning, coordination.

Nikos Vlassis is an assistant professor with the Division of Production Systems of the Department of Production Engineering and Management (DPEM) at the Technical University of Crete since 2007. He received his Ph.D. degree from the Technical University of Athens, Greece in 1998 and was an assistant professor with the University of Amsterdam, Netherlands until 2006. His current research interests include stochastic optimal control, unsupervised learning, and reinforcement learning. Vlassis has extensive experience with the RoboCup Simulation league and various distinctions with the UvA Trilearn robot soccer team, including the 1^{st} position at the RoboCup world championship (2003), three times 1^{st} position at the German Open tournament (2003, 2004, 2005), and the 1^{st} position at the American Open tournament (2003).

3 Team Members

Team Kouretes 2008 includes five members from two academic departments. The brackets indicate the main area each member is working on.

1. Daisy Chroni, undergraduate (DPEM) [Bipedal Stability]
2. Andreas Panakos, undergraduate (ECE) [Image Processing]
3. Alexandros Paraschos, undergraduate (ECE) [Bipedal Locomotion]
4. Georgios Pierris, undergraduate (ECE) [Bipedal Skills]
5. Efstathios Vafias, undergraduate (DPEM) [Bipedal Simulation (Team Leader for MSRS Soccer Challenge at Robocup 2008)]

4 Team overview

In this section we provide a high-level overview of the team and the software architecture of the MSRS-Nao robot. We assume that readers are familiar with the architecture of the Nao robot, the MSRS environment, and the generic architecture of the MSRS soccer player.

The main goal of MSRS-Kouretes is to create a competitive soccer playing team while improving the robot's movement and developing a more sophisticated player behavior.

Fig. 1. A snapshot of the MSRS simulation environment, featuring the Kouretes Nao's.

Control architecture: The robot needs to carry out its decision making based on its perception from the environment. The only sensor that the Simulated Nao Robot has is a camera, so the vision processing algorithms are our only way to find out what is going on in the field. The main logic loop is called every time we have an update from the camera. Each frame update triggers two different tasks; The first task uses the camera input to identify objects like the ball, the goals, and other Robot players. The second task involves the image processing results to determine the player behavior. The vision algorithm scans the image trying to locate specific colors (based on hue and saturation of each pixel) and then determine if the robot has spotted an object. Each time a recognizable object is observed, we calculate the relative angle between the body and the center of the spotted object and the approximate distance. After the vision part is over, the robot must determine its behavior with the data that is provided by the vision algorithm and based on its role (goalie or field player).

Goalie: The goalie's main purpose is to constantly follow the ball's movement, while trying to position itself inside the goal area in such a way so as to prevent the incoming ball. Our effort is to make the goalie able to calculate the speed of the incoming ball and stop it by falling down at the right time. Timing is crucial for this task, meaning that the more accurate we are on the ball speed tracking and calculation of the ball trajectory, the more efficient we will become in stopping opponent's shots.

Field player: The field player's role is a more complex one. In order to improve the field player's perception of the game it is important to find the best way to make the robot identify whether it needs to be in defensive or offensive mode and then act accordingly. This is accomplished by using its relative position between the ball, the opponent and the friendly goal. In defensive mode the robot will try to intercept the opponent by approaching a position between the opponent robot and the friendly goal. In offensive mode the robot is approaching the ball in an angle that will allow us to aim for the goal. Both offensive and defensive stance efficiency is based on the correct localization of the robots and the fast and accurate movement towards our desired position.

Team play: Until now, team play has not been an important part of the MSRS soccer league, but eventually in order to be able to reach human-like performance, the robots would have to develop a team like behavior. Our plan is to have each robot be able to communicate with each other (via DSS service operations in MSRS) in order to share information and form team strategies.

5 Motion parametrization

In this section we provide a description of some technical aspects related to the locomotion of the Nao.

In order to control the humanoid in a principled way we need to parameterize the motion primitives of the robot. This involves defining a set of trajectories $q_i(t)$, one for each joint angle q_i, which describe the evolution of q_i over time. Two standard approaches in the literature [1] involve splines, which allow modeling open-loop motions, and central pattern generators (CPGs), which can additionally model closed-loop motions. In our approach, the parameterized motion is open-loop, but instead of splines we use an expansion over trigonometric basis functions, as follows:

$$q_i(t) = a_0 + \sum_{k=1}^{K} a_k \cos(2\pi k t) + \sum_{k=1}^{K} b_k \sin(2\pi k t), \qquad (1)$$

where K is a fixed number (e.g., $K \approx 5$). The motivation for choosing trigonometric basis function is the fact that most of the motions of the humanoid are periodic, in which case a trigonometric basis set is a natural candidate. Additionally, a trigonometric expansion offers the possibility to easily learn an initial motion function from stored motion data, such as motion sequences designed using a motion editor. When these data are equally spaced in time (the typical case), trigonometric interpolation can be easily carried out by the discrete

Fourier transform. Towards this end, we analyzed the walk pattern provided with the motion library of the MSRS soccer player, and reproduced it using a trigonometric expansion. It turns out that with a number $5 < K < 10$ of basis functions we can reproduce the motion patterns with good accuracy.

The use of trigonometric functions (or any other fixed basis) allows us to parametrize the motion of the robot with a relatively small set of parameters (the coefficients a_k and b_k in eq. 1), which in turn allows for further optimization of the locomotion controller.

6 Static and dynamic balance

Another aspect of our work involves integrating a balancing module into motion control. This module ensures that the robot maintains its balance while carrying out some motion. This balancing module takes two forms: a static and a dynamic.

The static balancing module is used in cases where static balance of the robot is required, for example, when the robot tries to shoot the ball. The static balancing module ensures that the zero moment point (ZMP) [2] is within the support polygon of the ground foot of the robot, while the configuration of the robot changes continuously until the ball is hit. The latter is modeled as a constrained optimization problem, where the parameterized motion of the free foot (the one shooting the ball) plus the motion of several other joints (e.g. the arms) needed to maintain total posture balance, are computed in such a way that the ZMP stability constraint is not violated [3].

The dynamic balancing module mainly takes care of recovering from pushes and other external disturbances. Here, we use the accelerometers of the robot to detect abrupt acceleration caused by external forces, and then we estimate where the ZMP should be placed so that (static) stability is resumed. Our approach is related to the push recovery approach of Rebula et al. [4], where the balance recovery motion is learned from data instead of relying on the modeled robot physics.

7 Learning motions by reinforcement learning

A final aspect of our work involves the use of reinforcement learning (RL) for learning good motion functions for the various robot tasks (walking, rotating in place, ball kicking, etc.). Our approach is based on the natural actor-critic (NAC) framework [1]. This approach allows estimating (by running several trial-and-error episodes) the gradient of the value function of the parameterized motion policy as a function of the motion parameters (the quantities a_k and b_k in eq. 1). Then, by following this gradient we eventually reach a (local) optimum of the motion policy. The attractive properties of the NAC framework are its stability (convergence to a local optimum is guaranteed) and ease of implementation.

Alternatively, the problem of learning complex motion functions on a high-dimensional humanoid robot could be viewed as a multi-agent problem, whereby each agent controls a single joint of the robot and all controlling agents collaborate with each other towards a common goal. Recent independent research work by our team leaders has led to extensions of classic reinforcement learning algorithms to collaborative multi-agent learning where many agents learn to collaborate as a team [5, 6]. The scaling properties of these algorithms through exploitation of domain knowledge make them attractive for learning sophisticated motion skills for the Nao robot. While the large number of degrees of freedom of the robot imply a huge joint action space, this obstacle could be overcome by appropriate factorization of the representation on the basis of joint proximity on the robot body. Under such a learning scheme, the ankle joints of the left leg may need to "talk" to the knee joints of the same leg, but need not communicate directly with the ankle joints of the right leg. The resulting tree-like factorizations will make the above mentioned learning algorithms even more efficient as the required operations can be completed in polynomial time.

Acknowledgements

Team Kouretes would like to thank the administration of the Technical University of Crete for funding their travel to RoboCup 2008. The research efforts of the team were partially supported by the European Marie-Curie International Reintegration Grant MCIRG-CT-2006-044980 awarded to M. G. Lagoudakis.

References

1. Peters, J., Schaal, S.: Reinforcement learning of motor skills with policy gradients. Neural Networks (in press)
2. Vukobratovic, M., Juricic, D.: Contribution to the synthesis of biped gait. IEEE Transactions on Biomedical Engineering **16**(1) (1969) 1–6
3. J.J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue: Motion planning for humanoid robots. In Proc. 20th Int'l Symp. Robotics Research (ISRR'03), Italy, October 2003
4. Rebula, J., Canas, F., Pratt, J., Goswami, A.: Learning capture point for improved humanoid push recovery. In: IEEE-RAS 7th International Conference on Humanoid Robots, Pittsburgh, PA (2007)
5. Kok, J.R., Vlassis, N.: Collaborative multiagent reinforcement learning by payoff propagation. Journal of Machine Learning Research **7** (2006) 1789–1828
6. Guestrin, C., Lagoudakis, M.G., Parr, R.: Coordinated reinforcement learning. In: Proceedings of the Nineteenth International Conference on Machine Learning. (2002) 227–234

Κουρήτες
Kouretes
Technical University of Crete