Cerberus'08 Standard Platform League Team Qualification Document

H. Levent Akın[1], Çetin Meriçli[1], Tekin Meriçli[1], Barış Gökçe[1], Can Kavaklıoğlu[1], Ergin Özkucur[1] and Olcay T. Yıldız[2]

[1] Boğaziçi University, Dept. of Computer Engineering, Bebek, Istanbul, TURKEY
[2] Işık University, Dept. of Computer Engineering, Sile, Istanbul, TURKEY
{akin, cetin.mericli, tekin.mericli, sozbilir}@boun.edu.tr, eposta@cankavaklioglu.name.tr, eozkucur@gmail.com, olcaytaner@isikun.edu.tr

1 Introduction

"Cerberus" team made its debut in the 4-Legged league in RoboCup 2001 competition. This was the first international team participating in the league as a result of the joint research effort of Boğaziçi University (BU), Istanbul, Turkey and Technical University Sofia, Plovdiv branch (TUSP), Plovdiv, Bulgaria [1]. The team competed in Robocup since then, except Robocup 2004. Since 2004, Boğaziçi University is maintaining the team. In 2005, despite the fact that it was the only team competing with ERS-210s (not ERS210As), Cerberus won the first place in the technical challenges. In 2006, we have carried out our success with old ERS-210s to the more powerful ERS-7s by reaching the quarter finals. We lost only three games to the eventual first, third, and fourth place teams.

Boğaziçi University has a strong research group in AI. The introduction of Robocup as a unifying theme for different areas of study in autonomous robots has attracted many talented students and accelerated research efforts with more than 30 journal and conference papers. Currently, the department has teams both in Robocup four legged and rescue simulation leagues. Together with the introduction of the new humanoid robot Nao [2] as the new standard platform as a replacement for Aibo, the team now has a new challenge to make the robots play soccer in a more "humanlike" manner and the opportunity to test the methods developed for bipedal walking and control against other teams. We will be collaborating with Işık University. The variety of robotic platforms that the team has worked on so far also brought the necessity to develop a platform-independent robotics library [3], details of which will be provided in this document.

The rest of the document provides information about research done by our team in various fields of AI and robotics which we intend to apply to the new platform for playing soccer such as, vision, localization, locomotion, and planning and coordination/cooperation.

2 Software Architecture

We intend to use a modular approach which is based on the one that has been successfully applied on the Aibo platform. The modules that make up the architecture are:

– Vision
– Localization
– Locomotion
– Planning

In addition to the software running on the robots, we have already developed a sophisticated off line software called *Cerberus Station* which enables us to debug and test our algorithms, and helps us with labeling the images and several calibration operations. We will extend this platform to be used with Nao. We expect this process to be rather straightforward.

3 Vision

The vision module is responsible for information extraction from received camera frames. The process starts with receiving a camera frame and produces an egocentric world model consisting of a collection of visual percepts as shown in Figure 1.

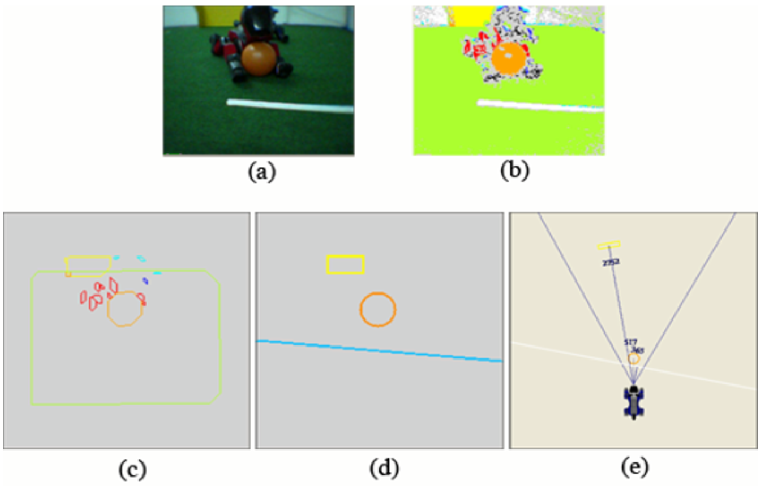

Fig. 1. Phases of image processing. a) Original image, b) Color classified image, c) Found blobs, d) Perceived objects e) Egocentric view.

Since the important objects on the field are color coded, the first step to take is to recognize these colors and detect corresponding objects (i.e. goals, beacons, and the ball). We use Generalized Regression Neural Network (GRNN) [4] for color generalization. In the ERS-7 case it was extended to cope with the radial distortion in the camera. If Nao's camera has similar distortions, these can be easily handled in a similar manner.

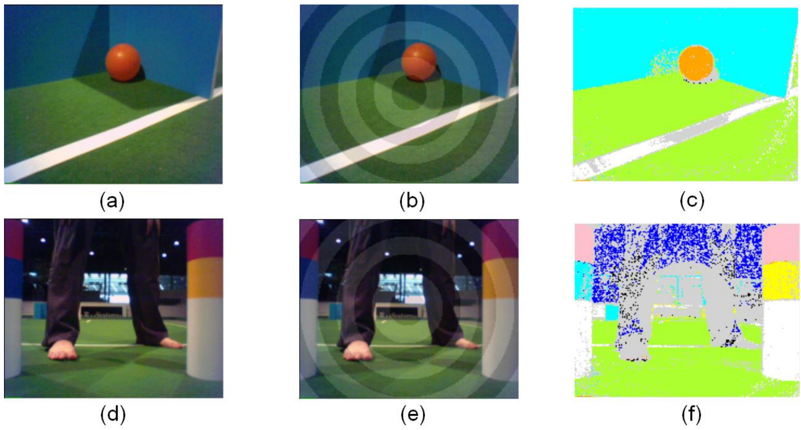

Fig. 2. The color classification. (a) and (d) Original Images, (b) and (e) Radius levels, (c) and (e) Classified images.

The first step in the training process is to collect a set of images from the robot's camera and hand label them with proper colors. Then, a GRNN is trained on the labeled data but instead of using only the Y, U, V triplet, an extra dimension indicating the euclidean distance of that pixel to the center of the image is also used. After the training phase, the network is simulated for the input space to generate a color lookup table for four bits (16 levels) of Y, six bits (64 levels) of U, six bits of V and three bits (eight levels) of the radius. The resulting color lookup table is very robust to luminance changes and it allows our vision system to work without using any kind of extra lights other than the standard ceiling fluorescents. With the introduction of distance component, the negative effect of the radial distortion is drastically reduced. According to our measurements, we can play reasonably at an illumination level of 150-200 lux. We expect our low level vision to perform better on Nao due to its relatively higher image quality than Aibos and allow us to play in natural lighting. The phases and result of our color classification method is shown in Figure 2 for an actual ERS-7 image.

After color classification is performed, the image is processed for obtaining blobs. We use an optimized region growing algorithm that performs both connected component finding and region building operations at the same time. This algorithm works nearly two times faster than the well known *RLE-Find connected components-build regions* approach.

Line perception is also an important part of the vision module, since it provides important information for the localization module. This component will be much more important with the removal of the beacons entirely in 2008. We use a Hough Transform based line perception algorithm. The sample images from line perception process are shown in Figure 3.

For detecting the goals that were started to be used in 2007, we use a scan-line based method with some improvements in the computational complexity of the process. Figure 4 shows a very challenging situation where the goal is almost completely covered

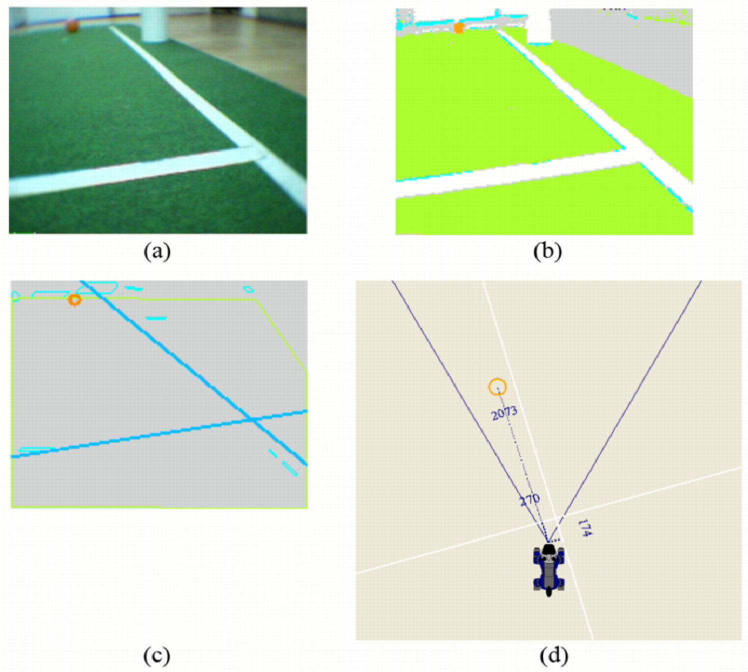

Fig. 3. Phases of Line Detection. a) Original image, b) Color classified image, c) Perceived lines e) Egocentric view.

with three robots; however, our goal detection approach is successful in detecting the goal even in such extreme cases.

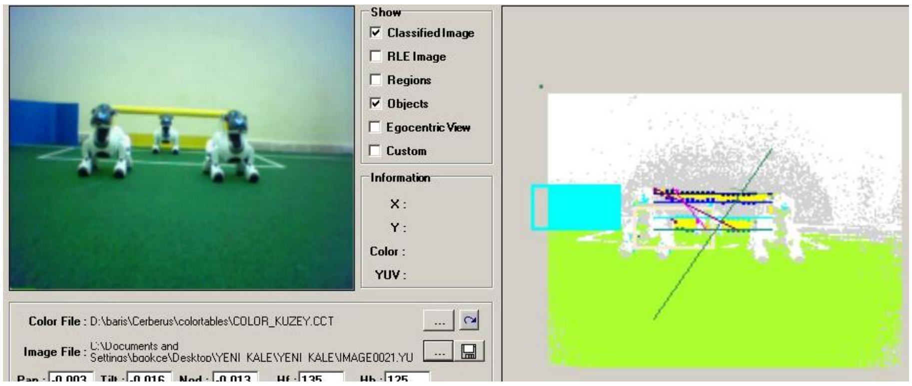

Fig. 4. Perception of the new goal on a highly occupied scene.

3.1 World Modeling

In our four legged robot team, each robot has an egocentric world model, which makes Markovian estimates about the robot's environment. The vision module provides instant and noisy data. However, we need smoother pose information and knowledge about our past perceptions in higher levels of planning. This is a state estimation problem and there are many applications in the literature for linear or non-linear models with different assumptions like Gaussian or distribution-free uncertainty approximation methods. In previous years Cerberus team used *MyEnvironment* [5] for this purpose. MyEnvironment keeps a window over the past observations and estimates the state by using statistics over the history. Starting from 2007, we switched to Kalman Filtering (KF) and Extended Kalman Filtering (EKF) methods [6] for tracking the objects around the robot.

First of all, we grouped objects in the environment as static and dynamic objects. For static objects, (beacons, goals, goal-bars, etc.) we used only KF, and for dynamic objects we used both KF and EKF. For static objects, our observation is $z = \{d, \theta\}$ where d and θ are distance and orientation which define polar coordinates of the object in egocentric coordinate frame. The state we are expecting is in the same format with our observation, which is $m = \{d, \theta\}$. The state transition model is linear and the states evolve according to the equations $m^1_{k+1} = Im_k$ and $z_k = Im_k$. For the dynamic objects, we want to estimate velocity components in vertical and horizontal axis in egocentric origin as well as their locations. While the observation vector is the same as static objects, the state vector is $m = \{x, y, dx, dy\}$, where x and y are coordinates, and dx and dy are velocity components. To estimate the state, an EKF is set up and used with the following observation model.

$$d = \sqrt{x^2 + y^2}$$
$$\theta = \arctan\left(y/x\right)$$

Our state transition model is as follows.

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ dx_{k+1} \\ dy_{k+1} \end{pmatrix} = \begin{pmatrix} 1\,0\,1\,0 \\ 0\,1\,0\,1 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ dx_k \\ dy_k \end{pmatrix}$$

The reason behind using two filters on the dynamic objects is that it is difficult to adjust the parameters of a single filter for estimating both the location and the velocity of the object at the same time. Therefore, we utilized one filter for estimating the location and the other filter for estimating the velocity.

The pose of the robot, the position, direction, and velocity of the ball, and the distance and orientation of the opponent goal are estimated through this egocentric world model. In our implementation of the market algorithm, which will be explained in Section 6, the robots share their egocentric world models through networking. Each robot fuses its own world model and the world models coming from its teammates, and forms a shared world model. The auction mechanism is executed by every robot using the information in the shared world model. Each robot calculates the costs for all robots for all tasks and runs the auctions. Since the shared world models are the same in all robots, they all make the same conclusion about the roles. This method is also safe compared to the one which involves a single auctioneer, from the point of view of single point failures.

4 Localization

Localization is one of our major research areas. Cerberus employs three different localization engines. The first engine is an in house developed localization module called Simple Localization (S-Loc) [7]. S-Loc is based on triangulation of the landmarks seen. Since it is unlikely to see more than two landmarks at a time in the current setup of the field, S-Loc keeps a history of the percepts seen and modifies the history according to the received odometry feedback. The perception update of S-Loc depends on the perception of landmarks and the previous pose estimate. Even if the initial pose estimate is incorrect, S-Loc acts as if the robot is kidnapped and it converges to the actual pose in a short period of time if the robot perceives enough number of landmarks during this period.

The second one is a vision-based Monte Carlo Localization with a set of practical extensions (X-MCL) [8]. The first extension to compensate for the errors in sensor readings is using inter-percept distance as a similarity measure in addition to the distances and orientations of individual percepts (static objects with known world frame coordinates on the field). Another extension is to use the number of perceived objects to adjust confidences of particles. The calculated confidence is reduced when the number of perceived objects is small and increased when the number of percepts is high. Since the overall confidence of a particle is calculated as the multiplication of likelihoods of individual perceptions, this adjustment prevents a particle from being assigned a smaller confidence value calculated from a cascade of highly confident perceptions in which case a single perception with lower confidence would have a higher confidence value. The third extension is related to the resampling phase. The number of particles in successor sample set is determined proportional to the last calculated confidence of the estimated pose. Finally, the size of the window in which the particles are spread is inversely proportional to the confidence of estimated pose.

The third engine is a novel contribution of our lab to the literature, called the Reverse Monte Carlo Localization (R-MCL) [9]. R-MCL is a self-localization method for global localization of autonomous mobile agents in the robotic soccer domain, which proposes a new approach to overcome the uncertainty in the sensors, environment, and the motion model. It is a hybrid method based on both Markov Localization (ML) and Monte Carlo Localization (MCL) where the ML module finds the region in which the robot should be and MCL predicts the geometrical location with high precision by selecting samples in this region (Figure 5). The method is very robust, and requires less computational power and memory compared to similar approaches, and it is accurate enough for high level decision making which is vital for robot soccer.

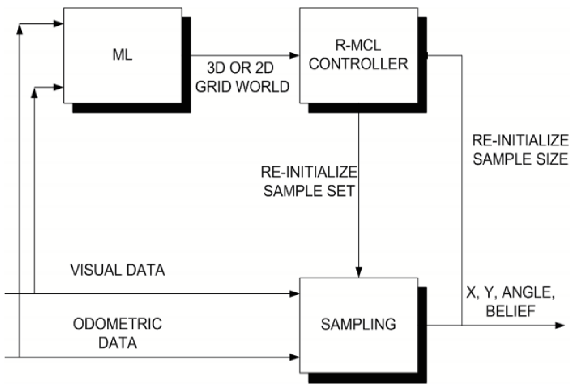

Fig. 5. R-MCL Working Schema.

5 Locomotion

The most important component of transition to the biped robotic platform will be the locomotion module. Fast and robust locomotion is a vital capability that a mobile robot should have, especially if the robot is expected to play soccer. Cerberus has a significant research background in different mobility configurations including wheeled and legged locomotion. Since the leagues Cerberus currently competes in require legged locomotion, the rest of this section will provide some information about our research in this field and the proposed extensions to biped locomotion.

5.1 Quadrupedal Walking

Almost all teams developed a different type of walk in the 4-legged league of RoboCup. The first parametric walking routine developed by rUNSWift 4-legged soccer team of UNSW [10] called *ParaWalk* has become the basis of most of the walking engines used in the league since it provides omnidirectional movement capability. In ParaWalk, the paws of the robot follow predefined loci and inverse kinematics is used to calculate the joint angles.

In 2005, we developed a rapid and stable parametric quadruped locomotion engine for Sony Aibo ERS-210 robots [11, 12]. The parameters of this engine are the number of intermediate points on the loci, the shape of the loci, and initial locations of the paws relative to the shoulder joint of each leg. The shape of the loci is a hermite curve approximated with an ellipse cut from below in some proportion. Optimization of these parameters is crucial in order to have a fast and stable locomotion; therefore, Genetic Algorithms (GA) is used as the optimization technique [12] in order to come up with the best parameter set. After some fine-tuning, the robots reached a forward walking speed of 310mm/s, which was the fastest walk achieved on ERS-210.

In 2006, we modified our motion engine for Aibo ERS-7 robots according to the hardware specifications, and the object-oriented design of our engine made this process easy and fast. The second step was to optimize the parameters of the engine for the dimensions of the new robot. First, we used GA for parameter optimization on the Webots simulator [13,14]. Then, we tried Evolutionary Strategies (ES) for optimization because of its advantage over GA in its ability to solve continuous parameter optimization problems [15, 16]. We implemented ES initially on the simulator as well in order to be able to make a deep search in the parameter space. After the convergence of the algorithm on the simulator, we implemented ES on the real robot to fine-tune the parameters for the carpet of the field [16].

5.2 Bipedal Walking

There has been a significant amount of research done on bipedal/humanlike walking. Most of the approaches are built on top of the Zero Moment Point (ZMP) concept [17] to control the stability of the robot. As opposed to the quadruped case, the robot is no longer in balance by default. In order to keep the robot upright and balanced, the trajectory that each foot follows changes over time; hence, the motion pattern must be generated in real-time. Angular momentum equation is used to guarantee that the generated trajectories are suitable for the dynamic stability. Dynamic stability of the robot can be measured by the distance of the ZMP to the boundaries of a predefined stability region.

Another algorithm that recently became popular is Passive Dynamic Walking (PDW) [18]. This algorithm tries to solve the problem in a totally different way. In PDW, passive fall is used as the main action during walking. While ZMP-based approaches try to keep balance continuously, PDW approach can be thought as continuous passive fall. The most important property of PDW is that the robot does not try to keep its balance. Balance is preserved only by changing the foot contact with the ground on time. In order to find the correct timing, this action can be examined as a cyclic motion. Using this approach, a more natural and efficient walking behavior can be obtained [19].

Our team started doing research on bipedal walking at the beginning of 2007. As in all the other robotic platforms that we have worked on so far, our goal with our humanoid robots has been to develop a platform-independent software architecture that will work both in the simulation environment and on the real robot. We started developing our algorithms on the simulator including a parametric bipedal walking engine; therefore, once we receive Nao, it will be relatively easy for us to run the codes on the robot after fine-tuning the parameters that we obtained on the simulator.

6 Planning and Cooperation

Planning is also a major research area of the Cerberus team. The soccer domain is a continuous environment, but the robots operate in discrete time steps. At each time step, the environment and the robots' own states change. The planner keeps track of those changes, and makes decisions about the new actions. Therefore, first of all, the main aim of the planner should be to model the environment appropriately and to update its status accordingly. Second, the planner should provide control inputs based on to this model. In robot soccer, the players should make both individual plans (single-agent planning) and plans as a team (multi-agent planning). The rest of this section provides the details of our approach to these two types of planning.

6.1 Single-Agent Planning

In the ERS-7 we used hand coded Finite State Automata based planners which generate the necessary behaviors [3]. We intend to do research on several approaches for generating planners in an autonomous fashion.

– *POMDP Based Approach:* Recently we extended POMDP algorithms to continuous domains [20] where we combine ART-2A [21], Kalman Filter, and Q-learning [22]. We used this approach to generate behaviors like *approach ball* (see Figure 6) and *kick the ball towards the goal* [23]. We will use this approach to generate more complex behaviors to be used by the planners.
– *Learning Based Approaches:* The robots learn both individual skills and collaboration, which emerge from very primitive behaviors. The action space is designed in such a way to obtain more complex behaviors by combining simple actions, similar to the subsumption architecture [24]. The simple actions include
  - going towards the opponent goal,
  - going towards the own goal,
  - going towards the ball and a specific target,

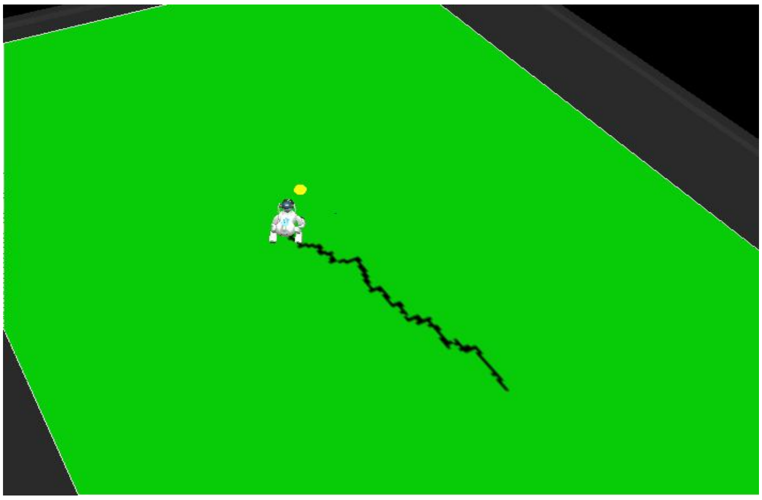

Fig. 6. The trajectory generated by ARKAQ algorithm.

– *Planning Based Approaches:* Due to lack of computational power and the slow nature of early classical AI planning algorithms, currently planning is not widely used in RoboCup. As a non-deterministic and partially observable system, earlier planners were not efficient or capable of presenting satisfactory results. We decided to asses the feasibily tof using such planners in real-time and developed a Fast Forward planner based approach [25] where the state was fully observable. It has recently been shown [26] that such planners can be extended to partially observable domains. The aim of this approach is to use latest planning algorithms in a non-deterministic and partially observable system such as RoboCup.

6.2 Multi-Agent Planning

Soccer is a team game, hence the decision of an individual soccer player is affected by all its teammates' decisions. This makes multi-agent planning an important research field for robotic soccer. We have been working on this problem by pursuing several different approaches.

– *Metric Definition and Validation:* Regardless of the planning algorithm to be used, one needs to have some quantitatively measured and informative metrics. We developed a set of novel metrics based on the positions of players and the ball over time and a novel statistical validation technique for testing whether a proposed metric is informative or not [27].
– *Market Based Approach:* We are using a market-driven multi-agent collaboration algorithm [28] for dynamic role assignment of the players. Once assigned, the agent starts making decisions according to its role until its role changes.

The market-driven method for multi-agent coordination [29] employs the properties of free markets, in which every individual tries to maximize its own profit, hence maximizing the system's total profit. The idea of market-driven method for multi-agent teams is based on the interactions between the agents in a distributed fashion for trading work, power, and information. The overall goal of the system is decomposed into smaller tasks and an auction is performed for each of these tasks. In each auction, the participant agents calculate their estimated cost for accomplishing that task and offer a price to the auctioneer. At the end of the auction, the bidder with the lowest offered price is given the right to execute the task and receives its revenue on behalf of the auctioneer. In order to implement this strategy, a revenue and a cost function must be defined. Then, the net profit can be calculated based on the difference between the values of the revenue and the cost functions.

We proposed this method for multi-robot collaboration in robot soccer domain [28, 30, 31]. We defined attacker, defender, and midfielder roles as tasks and we run an auction mechanism for each task. The robots calculate their costs for the tasks and participate in auctions. As an example, the cost function of being the attacker is calculated using the distance between agent and ball, distance between agent and goal, and clearance to the goal. The auctioning is done with the help of our shared world model structure.

– *DEC-POMDP Approach:* Partially Observable Markov Decision Process (POMDP) approach can model environments where the state is not fully observable and state transitions are probabilistic. Many studies have been made on this area in the last 30 years. POMDP models the environment from the point of view of a single agent. Decentralized POMDP (DEC-POMDP) which is an extended version of POMDP where there are multiple agents in the environment, is such an effort. We know that POMDP models usually give good results and they are applicable to real world situations. Therefore, DEC-POMDP is expected to be a good model for the multi-agent case. However, it has recently been shown that solving DEC-POMDP problems is NEXP-complete, while solving POMDP is PSPACE-complete [32]. Therefore, trying to obtain exact solutions for DEC-POMDP problems is not realistic. So, it is better to concentrate on finding near optimal approximate solutions. We have developed ES-based approximate solution approach for this purpose [33]. We are currently extending this to continuous states which will enable us to use them in realistic robotic applications like robot soccer.

7 Conclusion

Cerberus has been a very active team in the 4-legged league which is demonstrated both by the results and the quality as well as the quantity of publications which are the outcome of the research efforts. As demonstrated in this document the team is ready for the new standard league to apply its past theoretical and practical experience and to advance the state of the art in the fields of AI and robotics towards meeting the Robocup 2050 Challenge.

8 Acknowledgments

This work is supported by Boğaziçi University Research Fund through project 06HA102 and TUBITAK Project 106E172.

References

1. H. L. Akın. Managing an autonomous robot team: The cerberus team case study. *International Journal of Human-friendly Welfare Robotic Systems*, 6(2):35–40, 2005.
2. Aldebaran robotics - project nao.
http://www.aldebaran-robotics.com/pageProjetsNao.php.
3. Barış Gökçe Barış Kurt Can Kavaklıoğlu H. Levent Akın, Çetin Meriçli and Abdullah Akçe. Cerberus 2007 team description paper, 2007. Team Description Paper.
4. H. Schioler and U. Hartmann. Mapping neural network derived from the parzen window estimator. *Neural Networks*, 5:903–909, 1992.
5. Bulu¸Çelik Hatice Köse and Levent Akın. Comparison of localization methods for a robot soccer team. *International Journal of Advanced Robotic Systems*, pages 295–302, 2006.
6. Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.
7. B. Çelik Köse, H. and H. L. Akın. Comparison of localization methods for a robot soccer team. *International Journal of Advanced Robotic Systems*, 3(4):295–302, 2006.
8. T. Meriçli Ç. Mericli K. Kaplan, B. Çelik and H. L. Akın. Practical extensions to vision-based monte carlo localization methods for robot soccer domain. *RoboCup 2005: Robot Soccer World Cup IX, LNCS*, 4020:420–427, 2006.
9. H. Köse and H. L. Akın. The reverse monte carlo localization algorithm. *Robotics and Autonomous Systems*, 55(6):480–489, 2007.
10. UNSW 2003 Team Report.
http://www.cse.unsw.edu.au/robocup/report2003.pdf.
11. Cerberus 2005 Team Report.
http://robot.cmpe.boun.edu.tr/aibo/files/Cerberus05Report.pdf.
12. Tekin Meriçli. Developing a rapid and stable parametric quadruped locomotion for aibo robots, 2005. B.S. Thesis.
13. The webots mobile robotics simulation software.
http://www.cyberbotics.com/products/webots/.
14. Barış Gökçe. Simulation of genetic algorithm for gait optimization on webots and head kinematics. Technical report, Bogazici University, Department of Computer Engineering, 2006.
15. Cerberus 2006 Team Report.
http://robot.cmpe.boun.edu.tr/aibo/files/Cerberus06Report.pdf.
16. Barış Gökçe. Simulation of evolution strategies for gait optimization on webots and application on sony aibo ers-7 robots, 2006. B.S. Thesis.
17. M. Vukobratovic and B. Borovac. Zero-moment point-thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.
18. T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2):62, 1990.
19. M. Wisse and A.L. Schwab. First steps in passive dynamic walking. *International conference on Climbing and Walking Robots, Madrid*, 2004.
20. A. Sardağ and H. L. Akın. Kalman based finite state controller for partially observable domains. *International Journal of Advanced Robotic Systems*, 3(4):331–342, 2006.

21. S. Grossberg G.A. Carpenter and D. Rosen. Art 2-a: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, 4:493–504, 1991.
22. C. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, 1989.
23. D. Sezen. Implementation of continuous pomdp algorithms on autonomous robots. Master's thesis, Bogazici University, 2008.
24. R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal Of Robotics And Automation, RA-2*, pages 14–23, 1986.
25. Ü. D. Uluşar and H. L. Akın. Design and implementation of a real time planning system for autonomous robots. *2006 IEEE International Symposium on Industrial Electronics, ISIE'2006 - Montreal (Canada) - July 9-13, 2006*, 1:74–79, 2006.
26. E. Reisner U. Kuter, D. Nau and R. Goldman. Conditionalization: Adapting forward-chaining planners to partially observable environments. *ICAPS-2007*, 2007.
27. Çetin Meriçli. Developing a novel robust multi-agent task allocation algorithm for four-legged robot soccer domain, 2005. M.S. Thesis.
28. K. Kaplan Ç. Meriçli U. Tatlıdede Köse, H. and H. L. Akın. Market-driven multi-agent collaboration in robot soccer domain. *Cutting Edge Robotics*, pages 407–416, 2005.
29. M. B. Dias and A. Stenz. A market approach to multi-robot coordination. Technical report, 2001. CMU-RI-TR-01-26.
30. K. Kaplan H. Köse, Ç. Meriçli and H. L. Akın. All bids for one and one does for all: Market-driven multi-agent collaboration in robot soccer domain. *Computer and Information Sciences-ISCIS 2003, 18th International Symposium Antalya, Turkey, Proceedings LNCS 2869*, pages 529–536, 2003.
31. K. Kaplan H. Köse Tatlıdede, U. and H. L. Akın. Reinforcement learning for multi-agent coordination in robot soccer domain. *AAMAS05, Fifth European Workshop on Adaptive Agents and Multi-Agent Systems*, 2005.
32. S. D. S. Bernstein, S. Zilberstein and N. Immerman. The complexity of decentralized control of markov decision processes. *Proceedings Of The 16th Conference On Uncertainty in Artificial Intelligence*, 2000.
33. B. Eker and H. L. Akın. Solving dec-pomdp problems using evolution strategies. *Soft Computing*, 2007 (Submitted).