AllemaniACs@Home 2007 Team Description

Stefan Schiffer, Alexander Ferrein, and Gerhard Lakemeyer

Knowledge Based Systems Group
RWTH Aachen University,
Aachen, Germany
{schiffer, ferrein, gerhard}@cs.rwth-aachen.de

Abstract. This paper describes the scientific advances of the AllemaniACs team for the 2007 RoboCup@Home competitions. We present our low-level robot control software which allows us to perform reliable service robotics applications in the @Home league. Furthermore, we report on our high-level programming language providing a powerful framework for agent behavior specification.

1 Introduction

While in the RoboCup soccer leagues the complexity of the task lies in fast accessing the sensors, quick decision making, and cooperation, the challenge in the @Home league is to build a system which enables a robot to robustly and safely navigate through human populated home environments. Since the new RoboCup@Home league focuses on service robotics applications another challenge is that of human-robot interaction. Tasks like "follow & guide a human", "navigate within the environment", or "manipulate" are part of the @Home competition.

This means for one that the robot must be able to build an internal representation for arbitrary home environments. That is because the environment that the robot has to operate in for the competition is not known in advance. For another, the robot must be able to localize itself in this particular environment and it has to be able to navigate through it safely. This task surely demands for path planning and obstacle avoidance abilities. Our robot use a Monte Carlo approach with a laser range finder for localization. Furthermore, it employs an A^*-based collision avoidance algorithm and a path planner which ensures short paths between reachable points in the environment.

The high-level control is based on the language Readylog, a variant of the logic-based language GOLOG [1] which combines explicit agent programming as in imperative languages with the possibility to reasons about actions and their effects. In particular, we are interested in decision-theoretic planning in the Readylog framework which allows to generate optimal plans for complex tasks.

In the sequel we describe our hardware platform in Section 2. We present important aspects of our low-level control system in Section 3, before we sketch the high-level control language Readylog and give an example of a service robotics application of the AllemaniACs team from the 2004 RoboCup Technical Challenge in Section 4.

2 AllemaniACs @Home Robot

The mobile robot platform we use in the 2007 RoboCup@Home competitions is based upon the platform used in the AllemaniACS Mid-Size RoboCup Team. The robot has a size of 40 cm $\times$ 40 cm $\times$ 60 cm (Fig. 2). It is driven by a differential drive, the motors have a total power of 2.4 kW and are originally developed for electric wheel chairs. For power supply we have two 12 V lead-gel accumulators with 15 Ah each on-board. The battery power lasts for approximately one hour at full charge. This power provides us with a top speed of 3 m/s and 1000°/s at a total weight of approximately 60 kg. On-board we have two Pentium III PCs at 933 MHz running Linux, one equipped with a frame-grabber for a Sony EVI-D100P camera mounted on a pan/tilt unit. Our other sensor is a 360° laser range finder with a resolution of 1° at a frequency of 10 Hz. For communication a WLAN adapter based on IEEE 802.11b is installed.

Fig. 1: @Home robot platform

This hardware platform was initially designed for soccer playing, but with almost no modifications we can easily also use it for service robotics applications. We report on our transition from the Mid-Size to the @Home league in [2].

Since early 2007 we additionally have an anthropomorphic robotic arm called Katana6M180 which we intend to use for manipulation tasks. The Katana is equipped with six motors providing six degrees of freedom. In our current configuration the joint connecting the gripper is mounted in a straight fashion. All six axes are equipped with "Harmonic Drive" gears which allow precise movements and a high repeatability. The arm's weight is around 4 kg and it has a maximal payload of 500 g. The arm will be mounted on top of the mobile robot platform described above. Two provide the arm with the required power, we mounted two additional 12 V lead gel accumulators on the robot. Furthermore, we intend to install another camera on a higher level in order to be able to acquire additional sensory input. This could then be used for visual servoing of the arm.

Fig. 2: robot arm Katana

3 Low-Level Control

We use the 360° laser range finder as our main sensor for navigation, obstacle avoidance, and localization. In the following we describe the respective modules in more detail.

3.1 Collision Avoidance and Navigation

The collision avoidance module performs an A* search over an occupancy grid [3] generated from the laser scanner inputs. The robot is positioned in the middle (origin) of the grid. Next, the collision-free path from the current location to a given target point must be calculated. We perform an A* search from the robot's current location to the given target point. If the target point is located outside the grid range, we project the target point onto the border of the grid. To alleviate the search we extend the occupied cells by the size of our robot. Thus, the robot can be regarded as a mass point. The possible actions for the search are $A = \{N, S, W, E, NW, SW, ...\}$, i.e. the robot can move to any neighboring cell. To apply A* we need to provide a cost function and a heuristic function. The cost function is the Euclidean distance between grid cells, as heuristic function we use the Manhattan distance to the target point.

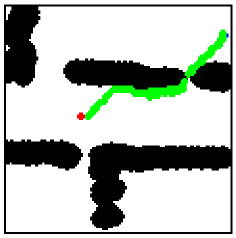

Fig. 3: Path to the target inside a room.

The path A* calculates (depicted in Fig. 3) must be translated into motor commands. Thus, we need a curve from which we can derive the appropriate commands sent to the motors. We approximate the steering commands by applying an A* search over the velocity space. This search yields appropriate translational and rotational velocities with which the robot drives to the given target point.

3.2 Localization

Our self-localization uses of the Monte Carlo Localization algorithm [4]. It works by approximating the position estimation by a set of weighted samples: $\mathbf{P}(l_t) \sim \{(l_{1,t}, w_{1,t}), \ldots, (l_{N,t}, w_{N,t})\} = \mathbf{S}_t$. Each sample represents one hypothesis for the pose of the robot. Roughly, the Monte Carlo Localization algorithm now chooses the most likely hypothesis given the previous estimate, the actual sensor input, the current motor commands, and a map of the environment.

To be able to localize robustly with the laser range finder we modified the Monte Carlo approach. To allow for the integration of a whole sweep from the LRF we use a heuristic perception model. With this we are able to localize with high accuracy in the RoboCup environment. The method is presented in detail in [5]. Our approach, which was inspired by the RoboCup setting, works also very well for indoor navigation even in large environments.

3.3 Object Classification

For localizing the robot in the environment we build an occupancy grid map of the environment. With this map, we are able to determine dynamic obstacles in the environment when new laser readings arrive: every object which is not represented in the map is assumed to be a dynamic obstacle. To be able to distinguish between different dynamic objects, we use the laser signature of the objects. In the soccer setting we are able to distinguish between our own robots and opponents, and even humans can be told apart.

3.4 Map Building Application

In order to be able to efficiently adapt to the frequent changes which are immanent in a home-like environment we developed a *semantic* map building application. It allows us to update the robot's world representation to the current situation very quickly. Our map builder uses a collection of semantically annotated objects that can be dragged and dropped to their specific location in a base-map. This simplifies the map building process to some few clicks. Semantic annotations include a signature of the object as seen by the laser range finder, the area to be used in the obstacle server, and a name along with some common aliases. If additionally a vision system is used one could also include sample pictures of the respective object. The particular information for each object have to be provided beforehand, e.g. the signature of an object as seen by the laser range finder has to be drawn or recorded and pictures need to be taken and associated with the object. The items in the different low-level data structures are inter-referenced by their name. This way, each module can refer to an object or place by its name in human terminology. Figure 3.4 shows a screenshot of the tool we developed for the map building task.

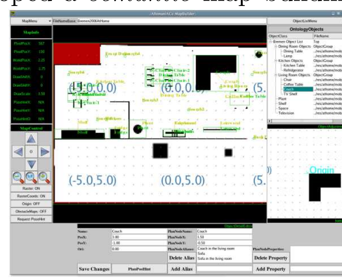

Fig. 4: map building tool

By providing the robot with semantically enriched objects it is able to make use of any particular part of the information associated with an object. Thus, interaction with humans in the environment can be achieved in a transparent fashion. For instance, when a human specifies a target location for the robot to reach, the name recognized by the speech recognition module is passed to the path planning module which in turn instructs the navigation module to drive to the associated coordinate within the map.

3.5 Human-Robot Interaction

In a natural human environment interaction between the robot and the human beings around it is an integral part of the challenges in the @Home league. Therefore, we realize communication facilities in terms of a speech recognition module to process human instructions, requests, and questions. To inform, answer, or ask the human for clarification on the current task we also provide a speech synthesis module which allows for generating spoken language.

For speech recognition we are using the Sphinx software system from Carnegie Mellon University. An overview of an early version of Sphinx is given in [6]. To model the interaction we realized a simple dialog system which is organized in a tree like structure. On the top most level the user can choose a specific task. Depending on the user's choice the robot offers further possibilities on the next level of the dialog hierarchy. For speech synthesis we make use of Festival. It was also developed at Carnegie Mellon University and features a simple interface to pass text which is then synthesized as speech. The initial Festival system is documented in [7].

(a) Part of the RoboCup 2004 Site in Lisbon

```
proc( pathPlan( Goal, H ),
      solve( while( not( mapNode = Goal ),
                    pickBest( child, childrenOf( mapNode ),
                    gotoMapNode( child ))),
             H, beAt( Goal ))).

function( beAt( Goal ), V,
          if( mapNode = Goal, V = 100, V = 0 )).

proc( gotoMapNode( Node ),
      [goto_global(ownNumber, nodeCoord( Node ), 1),
       playSound( Node )]).
```

(b) The READYLOG program for the 2004 challenge.

Fig. 5. Technical Challenge 2004

4 Readylog

For specifying our high-level control we use a variant of the logic-based high-level agent programming language GOLOG [1]. GOLOG is a language based on the situation calculus [8]. Over the past years many extensions like dealing with concurrency, exogenous and sensing action, a continuous changing world and probabilistic projections (simulation) [9,10,11] made GOLOG an expressive robot programming language. We integrated those features in our READYLOG interpreter [12]. For the decision making, we further integrated a planning module into GOLOG which chooses the best action to perform by solving a Markov Decision Process (MDP). For examples of how a multi-agent plans for the robotic soccer domain can be formulated in READYLOG we refer to [**?**,12]. For further extensions of the READYLOG interpreter we refer to [13,14].

In 2005 we developed a qualitative abstraction of the world model for the MID-SIZE domain [15,16]. The qualitative world model is integrated in the READYLOG language and used for abstract planning. The qualitative world model provides abstractions for positional information such as *left* or *right* as well as higher-level concepts like that of reachability which is fundamental in soccer. The qualitative spatial data provided by this world model are based on human cognition. Thus, they render useful especially when it comes to human-robot interaction since the robot can handle information which originate from human language more easily.

In the ROBOCUP MID-SIZE Technical Challenge 2004 we already presented a service robotics application: the robot was to drive autonomously to one particular soccer field which was chosen by one of the referees. The robot calculated the shortest way to the field, announcing historic sights of the exhibition hall like the stand of the fields or the pillars of the hall on the way. Fig. 5(a) shows the occupancy map of the rear part of the exhibition hall. In the upper part you could detect two of the MID-SIZE fields. The high-level control program for the challenge is shown in Fig. 5(b).

References

1. Levesque, H.J., Reiter, R., Lesperance, Y., Lin, F., Scherl, R.B.: GOLOG: A logic programming language for dynamic domains. Journal of Logic Programming **31** (1997) 59–83
2. Schiffer, S., Ferrein, A., Lakemeyer, G.: Football is coming Home. In: Proc. of International Symposium on Practical Cognitive Agents and Robots (PCAR'06), University of Western Australia Press (2006)
3. Moravec, H., Elfes, A.: High resolution maps from wide angular sensors. In: Proc. of the IEEE International Conference On Robotics and Automation (ICRA). (1985) 116–121
4. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte Carlo localization for mobile robots. In: Proc. of the Int. Conf. on Robotics and Automation (ICRA). (1999)
5. Strack, A., Ferrein, A., Lakemeyer, G.: Laser-Based Localization with Sparse Landmarks. In: Proc. RoboCup Symposium 2005. (2005)
6. Huang, X., Alleva, F., Hon, H.W., Hwang, M.Y., Rosenfeld, R.: The SPHINX-II speech recognition system: an overview. Computer Speech and Language **7** (1993) 137–148
7. Black, A.W., Taylor, P.A.: The Festival Speech Synthesis System: System documentation. Technical Report HCRC/TR-83, Human Communciation Research Centre, University of Edinburgh, Scotland, UK (1997)
8. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press (2001)
9. Giacomo, G.D., Lésperance, Y., Levesque, H.J.: ConGolog, A concurrent programming language based on situation calculus. Artificial Intelligence **121** (2000)
10. Grosskreutz, H., Lakemeyer, G.: cc-Golog: Towards more realistic logic-based robot controllers. In: Proc. AAAI'00, AAAI Press (2000) 476–482
11. Grosskreutz, H.: Probabilistic projection and belief update in the pGolog framework. In: Second International Cognitive Robotics Workshop. (2000)
12. Dylla, F., Ferrein, A., Lakemeyer, G.: Specifying multirobot coordination in ICPGolog – from simulation towards real robots. In: Proc. Physical Agents WS on IJCAI'03. (2003)
13. Ferrein, A., Fritz, C., Lakemeyer, G.: On-line decision-theoretic golog for unpredictable domains. In: Proc. of 27th German Conference on AI. (2004)
14. Ferrein, A., Fritz, C., Lakemeyer, G.: Extending DTGolog with Options. In: Proc. IJCAI'03. (2003)
15. Schiffer, S., Ferrein, A., Lakemeyer, G.: Qualitative World Models for Soccer Robots. In: Qualitative Constraint Calculi, Workshop at KI 2006. (2006) 3–14
16. Schiffer, S.: A Qualitative Worldmodel for Autonomous Soccer Agents in the ReadyLog Framework. Master's thesis, Knowledge-based Systems Group, Computer Science Department, RWTH Aachen University, Aachen Germany (2005)