Skill Acquisition and Use for a Dynamically-Balancing Soccer Robot

Brett Browning, Ling Xu and **Manuela Veloso**
Carnegie Mellon University
School of Computer Science
5000 Forbes Avenue, Pittsburgh, USA
brettb,mmv@cs.cmu.edu, lingx@andrew.cmu.edu [*]

Abstract

Dynamically-balancing robots have recently been made available by Segway LLC, in the form of the Segway RMP (Robot Mobility Platform). We have addressed the challenge of using these RMP robots to play soccer, building up upon our extensive previous work in this multi-robot research domain. In this paper, we make three contributions. First, we present a new domain, called Segway Soccer, for investigating the coordination of dynamically formed, mixed human-robot teams within the realm of a team task that requires real-time decision making and response. Segway Soccer is a game of soccer between two teams consisting of both Segway riding humans and Segway RMPs. We believe Segway Soccer is the first game involving both humans and robots in cooperative roles and with similar capabilities. In conjunction with this new domain, we present our work towards developing a soccer playing robot using the RMP platform with vision as its primary sensor. Our third contribution is that of skill acquisition from a human teacher, where the learned skill is then used seamlessly during robot execution as part of its control hierarchy. Skill acquisition and use addresses the challenge of rapidly developing the low-level actions that are environment dependent and are not transferable across robots.

Introduction

There has been considerable research into both human-robot interaction (Nicolescu & Mataric 2003), and multi-agent teams (Dias & Stentz 2002; Ferraresso *et al.* 2001; Kiat *et al.* 2001). Additionally, since the inception of RoboCup robot soccer (Asada *et al.* 2003), there has been considerable research into robot teams operating in adversarial environments. To our knowledge, however, there has been no work yet that combines these attributes; namely, to examine human-robot interaction within an adversarial, multi-robot setting where humans and robots are team members with similar capabilities and no clear role hierarchy.

We are developing a new game, which we call Segway Soccer, that aims to fill this void. Segway soccer is a game that requires mixed teams of humans and robots to cooperate to achieve the maximum reward in an adversarial task. To ensure interesting cooperation, both humans and robots are equipped with similar capabilities. We achieve this difficult task by requiring that both humans and robots use the same drive platform - the Segway platform developed by Segway LLC (Figure 1).

Our goal is to create a task that requires advanced robot intelligence, combined with robust human-robot interaction skills. We hope to extend the powerful aspects of RoboCup - competition, an adversarial domain requiring fast decisions, and a well understood task - to incorporate human-robot interaction. The need for this new domain lies in the lack of study for human-robot interaction where decisions need to be made quickly. As robots become more integrated into society, they will inevitably have to interact with humans and/or legacy robots in complex tasks. For some of these tasks, decisions may need to be made quickly and roles of both humans and robots may not be clearly defined a priori.

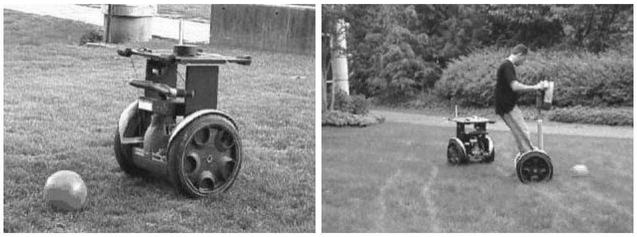

Figure 1: The Segway RMP (left and right) and HT (right) platforms developed by Segway LLC (http://www.segway.com).

In this paper, we describe our work towards developing a robot capable of participating in Segway soccer. As this new domain is set in the outdoors, compensating for variable lighting conditions and less structured environments, but still retaining the ability to make and act on decisions quickly is a challenging task. We describe our initial solutions to meet this challenge.

[*] This research was sponsored by the United States Army under Grant No. DABT63-99-1-0013. The content of the information in this publication does not necessarily reflect the position or the policy of the Defense Advanced Research Projects Agency (DARPA), the US Army or the US Government, and no official endorsement should be inferred.

The format of the paper is as follows. In the following section we describe the specifics of Segway soccer; its rules, structure, goals, and challenges. Following this we describe how to extend a Segway RMP to become a robot capable of playing soccer with vision as its primary sensor. We then focus on our third contribution, skill acquisition and use, leading to our empirical result, conclusions and future work.

Segway Soccer

In this section, we concretely describe the game of Segway soccer. We begin by describing the unique features of the Segway platform that make the concept of human-robot teamwork feasible in a complex game such as soccer, before progressing to the game itself.

Segway as a Platform for Robotics Research

The Segway platform, invented by Dean Kamen, is unique due to its combination of wheel actuators and dynamic balancing. For the robot version this imbues the robot with the novel characteristics of a fast platform, able to reach speeds of $3.5m.s^{-1}$ and travel long ranges (on the order of kilometers), able to carry significant payloads, able to navigate in relatively tight spaces for its size, and provides the opportunity to mount sensors at a height comparable to human eye level. As it is derived from a commercial product, it is reliable and robust to damage. Finally, dynamic balancing offers a number of unique properties including low-pass filtering of perturbations by non-flat terrain, as well as a moderate level of compliance with collisions.

From the perspective of human-robot interaction, the Segway is unique in that its human and robot versions have identical physical capabilities. The only difference resides in different perception and cognition capabilities of humans versus robots. For exploring human-robot interaction in cooperative teams operating in adversarial tasks requiring real-time responses, this property is essential.

For control, the RMP presents a high-speed serial interface, using the automotive industry standard CAN bus. Although essentially identical to the human-ridable Segway HT, the robot version has a modified control algorithm enabling a connected laptop to send desired forward and rotational velocity commands. The RMP's onboard algorithms accelerate or decelerate the robot to achieve these commands. Finally, the robot provides feedback on its current state including full angular pose and velocity information. Physically, the robot appears as a table on wheels due to the addition of a large mass of approximately 50lbs at a height of about 50cm from the robot wheel base. This mass raises the robot's center of gravity thereby enabling the RMP to balance with a control loop operating at a realizable frequency.

The Game

We have developed the domain of Segway soccer for investigating human-robot interaction. Segway soccer is a game between two teams playing on a grass field in an outdoor environment with an orange, size 5 soccer ball. Teams can consist of humans, robots, or a mix of humans and robots. Figure 2 shows the field structure. The field consists of a grass surface in an outdoor environment, where the size is scaled as a function of the number of team members so that with 11 players on each team it is a full-sized soccer field. Colored tubular markers are placed around the field to indicate the field boundary and goal locations. A human referee maintains control of the game and transmits signals verbally, for human consumption, and via wireless communication for the robots. The latter is achieved via an assistant referee armed with a laptop and a wireless network, an approach developed in robot soccer (Asada *et al.* 2003). Team members may be robots, humans, or robots and humans. In all cases, the Segway platform is used to ensure each team member has identical physical capabilities. Both humans and robots wear colored markers to allow easy team identification.

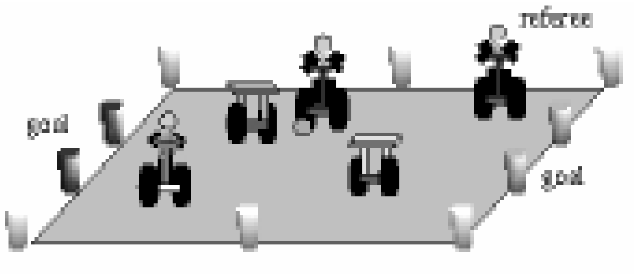

Figure 2: The Segway field. Teams consist of humans, robots, or robots and humans using the Segway platform, and an orange size 5 soccer ball.

As both Segway HT's and RMP's carry considerable mass, and are able to reach considerable speed, safety is a primary concern. To address this problem, the game follows a flow more familiar to Ultimate Frisbee[1]. When play begins, ball possession is decided with a coin toss. Thereafter, players gain possession based on proximity to the ball when it is ”free”. Once a player obtains possession, opponents must remain at least $3m$ away to prevent any unnecessary contact. Players are not allowed to move with the ball (i.e. dribble), and instead must pass the ball to one another for the team to maintain possession. A time limit on possession enforces how long a single player can retain the ball before passing to a teammate. When the ball is passed, the first player on any team to come within a specific distance of the ball when it comes to rest will gain possession. The same player cannot re-acquire possession of the ball until after another player has obtained possession. Possession is also changed if the ball is kicked out of bounds or if a goal is scored. Although primarily a safety measure, this rule also ensures that players must pass the ball to advance. As a direct consequence teamwork, rather than purely single robot skills, becomes essential. The goal of exploring intelligent teamwork is therefore achieved.

Although the rules as defined allow for a multi-agent, adversarial game to be played, they do not necessarily enforce human-robot interaction. If, for example, humans prove considerably more capable than their robot teammates, or vice-versa, one class of team members will dominate pos-

[1] Rules for Ultimate Frisbee can be found at http://www.upa.org

session leading to little human-robot interaction opportunities. Either case is undesirable. Our solution to this problem is to require that both a human and a robot be part of the sequence of passes leading to a goal score. How effective this solution is in the long term under competition conditions, remains to be seen.

Developing a Segway Soccer Player

We now describe our work to develop a Segway RMP robot base capable of playing Segway soccer. As with any autonomous robot operating in a dynamic world, one must develop a *complete* system involving perception, cognition, and action inter-operating seamlessly with minimal response latency. In this section, we detail our contributions to achieving this goal which builds upon our prior work in this area.

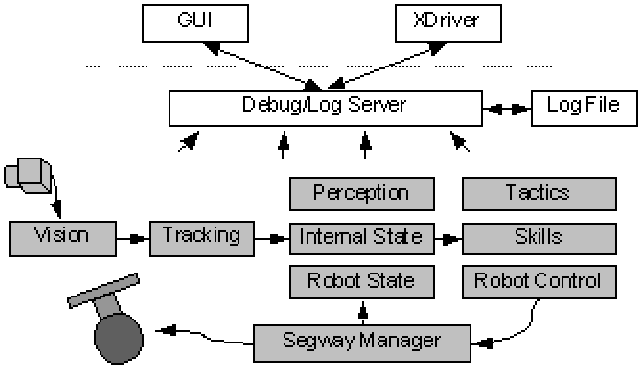

Figure 3: The control hierarchy used for the robot. The gray modules are the perception-cognition-action part of the system. The white are development infrastructure aids. Xdriver is the tele-operation program.

Figure 3 shows the complete control architecture for the RMP. The gray boxes show the main processing path that makes up perception, cognition, and action. In an environment occupied by other fast moving agents, the ability to perceive and respond to situational changes in minimum time is essential. Hence, it is critical to overall robot performance that the control loop formed by gray modules operate at full frame rate with minimum latency. The white boxes show the supporting development environment, which although not critical during execution play a central role in the pace of the development cycle and therefore in the robustness of the result. We now describe each major component and its role in the overall hierarchy, namely; perception, skills and tactics.

Perception: Vision and Tracking

For environments like those the RMP operates in, there are few sensors that can compete with color vision for low cost, compact size, high information volume and throughput, and relatively low latency. Thus, our chosen path is decidedly vision centric where a color camera provides each robot with its only external sensor. Given the range of environments that the robot operates in – outdoors on grass fields, or indoors when the weather is poor – perception quickly becomes an overriding challenge. That is, to develop *fast* vision algorithms that provide the robot with timely, but relatively noise free information that is robust to variations in lighting intensity. To complicate issues, only a fraction of the available processing power is dedicated to vision as the remainder must be used for tracking, modeling, behaviors, navigation and motion control. Achieving these goals is one of the key challenges to be addressed for using the RMP. To our knowledge, there are no vision algorithms that offer all of these features.

A number of fast, color-based algorithms and freely available source libraries have been developed for static lighting conditions (e.g. (Bruce, Balch, & Veloso 2000)). However, these libraries do not as yet extend to variable, or changing lighting conditions. Our approach uses region growing (Adams & Bischof 1994), where seeds are first provided from any regions that found in the previous frame that were large enough. The remainder of the seeds are chosen uniformly. Each region is grown using a fixed homogeneity constraint based on the distance of the new color pixel in YUV space from the average color of the region grown thus far. That is, if the new pixel has color $c_j = (y_j, u_j, v_j)^T$, it is added to the region R_i if it is a neighbor of an existing pixel in the region, $c_j \in \mathcal{N}(c_k),\ c_k \in R_i$, and it is sufficiently close to the region mean, $(y_j - \hat{y}_i < \tau_y) \wedge (u_j - \hat{u}_i < \tau_u) \wedge (v_j - \hat{v}_i < \tau_v)$. The region mean is updated after each pixel addition and has the value $\hat{c}_i = |R_i|^{-1} \sum_{R_i} c_j$.

Once regions are identified and the summary statistics for each region are calculated, high level vision processing begins by identifying the regions that are potential objects of interest: the ball, field markers, teammates or opponents. Regions that are close to the color of interest, using a Euclidean distance metric and threshold, are labelled accordingly. That is for prototype p: $R^p = \{R_i |\ \|\hat{c}_i - c^p\| < \tau^p\}$. A region may be labeled as belonging to more than one class type. For each object of interest, the regions are then based on objects' expected geometric properties as in our earlier work (Lenser, Bruce, & Veloso 2001). Figure 4 shows an example of the vision processing and its output.

The goal of vision is to provide as many valid estimates of objects as possible (i.e. a low false-positive rate). Estimates of the global positions of the obstacles is derived using a lens distortion model, a pin-hole projective model for the camera, and knowledge of the robot's respective tilt angles. Tracking then fuses this information to track the most interesting objects of relevance to the robot. At the time of writing our current multi-hypothesis tracker is still under active development. Ongoing work is focused on developing a true probabilistic multi-hypothesis tracker (Bar-Shalom 1990).

Robot Control Hierarchy

In our previous work we developed and thoroughly validated a hierarchical, behavior based control system called Skills-Tactics-Plays (STP) (Bruce *et al.* 2003 to appear) for adaptive multi-robot control in adversarial, highly dynamic environments. The architecture consists of Skills for low-level control policies, Tactics to encapsulate a complete single robot behavior, and Plays for team coordination. We have

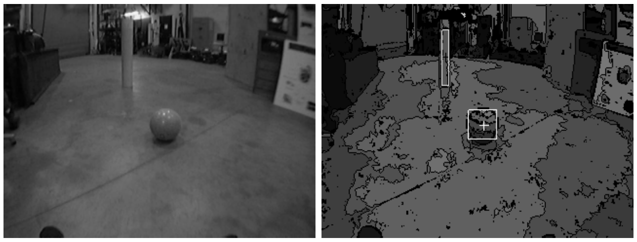

Figure 4: The left image shows a raw image in an indoor scene, the right the resulting regions, with the identified ball region and yellow marker outlined with a bounding box.

applied the STP control hierarchy to the Segway problem. Here we focus on skills and tactics that encapsulate single robot behavior.

Robot Control At the lowest level of the hierarchy are motion control and obstacle free navigation, key components to any mobile robot. These modules are adapted versions of our prior work, (Bruce *et al.* 2003 to appear) and (Bruce & Veloso 2002), respectively, and will not be discussed in detail here. Above motion control are the skills, the building blocks of individual behavior.

Skills Each skill is a focused control policy for carrying out complex actions over some limited set of world states. For Segway soccer an example skill is the action to actually kick the ball, or to position behind a ball in order to kick it towards a target.

Tactics A tactic encapsulates a complete single robot behavior. An example tactic includes shooting the ball on goal, receiving the ball from a teammate, or defending the goal.

Plays A play encapsulates team behavior by encoding a sequence of tactics to be carried out by each team member (Bowling, Browning, & Veloso 2004 in press). Plays are beyond the scope of this paper.

Skills are the action primitives for tactics, thus a tactic consists of instantiating a sequence of skills to execute, in other words a finite state machine, where the sequence of execution depends upon how the perceived world state changes. Tactics affect the world by instantiating skills in sequence to execute and by passing each executing skill parameters derived from the world state to affect its operation as it executes. For example to shoot the ball at the goal, the shoot tactic executes a sequence of skills such as gotoBall, positionForKick, and when aimed at the target the final kick skill. Finally, following the usual behavior based approach (Arkin 1998), tactics and skills execute in parallel at frame rate (30Hz).

With only minor variations all of the tactics employed on the RMP were developed previously in (Bruce *et al.* 2003 to appear). At this high-level of behavior, the tactics evaluate the world and determine target points for kicking or moving. As most of the details of each action are contained in the underlying skills, we have found that the tactics transfer effectively from one platform to another completely different platform but for a very similar task. In contrast, the underlying skills are highly dependent up on the hardware and do not transition effectively. To avoid the necessity of redeveloping a new skill set, we followed the novel approach of developing skill acquisition systems for rapidly acquiring new skills through training by a human operator.

Skill Acquisition and Use

Within the STP framework, skills form the closest interface to the physical robot hardware. Each skill, therefore, is highly dependent upon the physical properties of the robot and its environment and plays a major role in determining overall robot performance. The dependency of skills on the physical properties of robot and environment mean that skills rarely transfer well from one environment to the next, or from one robot platform to another, where deviations are the norm. When one considers that in practice, developing a skill requires considerable hand tweaking of control parameters, a notoriously error prone and time intensive operation, it seems apparent that some form of automated skill acquisition is needed.

One interesting approach to skill acquisition is reinforcement learning (e.g. (Bagnell *et al.* 2003)). Although there have been significant advancements in reinforcement learning, there are still a number of issues that must be address in order to learn effective robot control where *it could be used as a sub-component of an existing control hierarchy*. The primary limitation is that of long training time. Even the most advanced algorithms still take a considerable amount of time to achieve even moderate levels of performance.

Approach

We have taken a different approach to skill acquisition, where skills commands are generated by generalizing from example trajectories provided by a human operator using tele-operation. There have been a number of examples in the literature (e.g. (Schaal & Atkeson 1998; Billard & Mataric 2001) and (Nicolescu & Mataric 2003)) where tele-operation or learning from observation has been effectively used to learn to execute a task thus motivating our approach. Our particular approach is inspired by two observations. First, it is usually relatively easy to tele-operate a wheeled robot, even a dynamically balancing one, through the approximate sequence of motions required to execute a skill. Second, skills developed by hand are often a direct, if complex, function mapping from sensory state to output commands. Our key assumption is that the commands given by the human operator are noisy samples from some fixed, but unknown function of the world state. The goal of skill acquisition is therefore to provide estimates of this function for world states experienced by the robot based on the provided samples.

There are numerous function approximation techniques that are available. However, we have focused our investigations on locally weighted regression (LWR) (Cleveland & Loader 1995). LWR provides robust function estimation using a minimal amount of training data, is able to generalize by interpolation across unseen inputs, and is amenable to fast, efficient implementations. All of these criteria are of importance to skill acquisition in the given setting. Additionally, LWR has a long, successful history of use in robot control problems (Atkeson, Moore, & Schaal 1997; Schaal & Atkeson 1998).

Our approach works in two phases; recording and playback. During the recording phase, a human operator guides the robot through a sequence motions by providing a sequence of commanded actions $a_i = \left(a_i^1, a_i^2, ..., a_i^k\right)^T$. A special purpose recording tactic stores the sent command a_i, as well as the corresponding relevant world state $x_i = \left(x_i^1, ..., x_i^d\right)^T$ for later recall. During playback, the skill uses these data-points to approximate the actions of the recording phase. The key assumption is that the recorded actions are sampled from some unknown function so that $a_i = f(x_i)$. During playback the skill approximates this function, using its samples, using LWR, such that i.e. $a(t) = \hat{f}(x(t))$ as the robot moves in the world. Concretely, we have:

$$a(t) = \hat{f}\left(x(t)\right) = \frac{\sum_i K\left(x(t), x_i\right) \cdot a_i}{\sum_i K\left(x(t), x_i\right)} \tag{1}$$

where $K(\cdot, \cdot)$ is a kernel function, which in practice is a Gaussian given by:

$$K\left(x(t), x_i\right) = e^{\frac{-(x(t)-x_i)^2}{2 \cdot h^2}} \tag{2}$$

Implementation

There are three issues to implementing the LWR function approximation. The first is to provide for fast function approximation. Following the usual approach, we store the x_i's in a Kd-tree for fast nearest neighbor search (Atkeson, Moore, & Schaal 1997). Additionally, we limit the horizon of the search to $H_{max} = sqrt\frac{2h^2}{lnK_{min}}$, for some predefined K_{min}, thereby limiting the extent of the evaluations of 1. Secondly, one must choose the so-called bandwidth parameter, h. We chose a global value of h by hand. An alternative approach would be to use cross-validation, or a locally varying bandwidth parameter (Cleveland & Loader 1995). Although there are some arguments for the usefulness of the latter, in our current work it proved unnecessary. Finally, while LWR works well for interpolation, it is well recognized that it becomes unpredictable for extrapolation beyond the underlying data set, i.e. outside of the convex hull formed by the x_i's. For robot control this can cause serious limitations. However, given that skills are by definition only defined over a sub-set of the state space, say S_{skill}, provided this set of states is a sub-set of the convex hull formed by the x_i's no difficulties should occur. This is achieved by ensuring the skill training data covers S_{skill}. As a secondary safety measure, should the normalization factor in 1 be too small the output is set to the null vector and warning notifications are given.

Experimental Results

All of the work described here has been fully implemented on the RMP platform. Moreover, all components of the system are fully integrated allowing execution at the full frame rate of 30Hz. Video demonstrations of the system in execution are available at *http://www.cs.cmu.edu/ robosoccer/segway*. While many sub-components have been built upon our previous extensive experiences with robot teams operating in dynamic environments, it was not possible to build upon our existing skill libraries developed for other platforms. The RMP's unique dynamic balancing means that short-term motions are distinctly different from a comparable differential drive robot. Thus, we have a practical need for effective skill acquisition and use.

All of the low-level skills operating on the RMP, except for the most trivial, utilize skill acquisition. However, to properly evaluate the performance of skill acquisition as described here, we made use of our small-size robots (Bruce *et al.* 2003 to appear) and high-fidelity simulator UberSim (Browning & Tryzelaar 2003), where accurate ground truth information is easily available. The robots were differential drive robots, but are statically balanced. We trained a skill using a hand-coded policy $\pi\left(x(t)\right)$ that drives the robot towards the ball. A hand coded policy was used instead of tele-operation in order to get a qualitative evaluation in execution performance. The skill was acquired using the training gathered training data and then evaluated on a previously unseen evaluation point that was inside the convex hull of the previously training trajectories. We then compared the performance of the acquired skill against the hand coded policy evaluated from the same location (see table 1).

```
Training
1. for each training point $x_j^{train}$
3.     Execute $a(t) = \pi^{hc}(x(t))$ until goal reached
4.     Record $(x(t), a(t))$ for learning

Testing
1. Load data points $x_i, a_i$
2. for each testing point $x_j^{test}$
3.     Execute $a(t) = \hat{f}(x(t))$ until goal reached
4.     Record $x^{acq}(t)$ for evaluation

Evaluation
1. for each testing point $x_j^{test}$
2.     Execute $a(t) = \pi(x(t))$ until goal reached
3.     Record $x^{hc}(t)$ for evaluation
```

Table 1: The evaluation procedure.

For learning, the state was encoded as the relative location of the ball as $x(t) = \left(b_x(t), b_y(t)\right)^T$. The commands were encoded as the forward and rotational speed, $a(t) = (v, \omega)^T$. The hand coded policy was used from existing, publicly available algorithms (Bruce *et al.* 2003 to appear). Figure 5 shows the comparison between the robot trajectories for the acquired skill ($x^{acq}(t)$) and the hand coded policy ($x^{hc}(t)$) for different starting conditions, with the ball located at $(0, 0)$. Twelve trajectories distributed over 4 grid points points with starting angles of 0, -90, and 90 degrees, were used to train the skill. Even with only a few examples, the acquired skill is able to show the appropriate operation

Evaluation	Run 1	Run 2	Run 3	Run 4	Run 5
Hand coded	0.63	0.87	0.63	0.87	0.97
Acquired	0.70	0.77	0.60	0.97	1.03

Table 2: Comparison of the run time (in seconds).

with only minor amounts of error. Table 2 shows the difference in time to reach the goal for 5 of these runs.

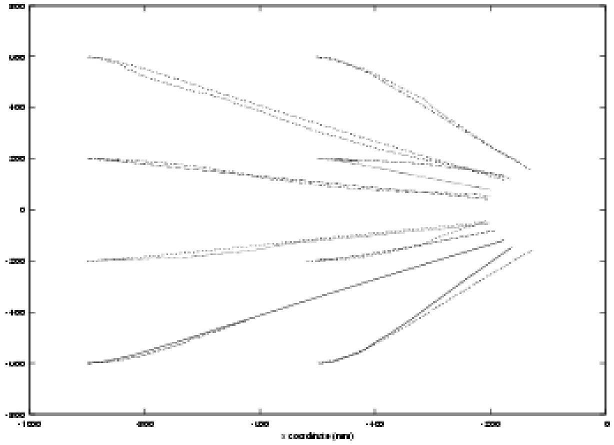

Figure 5: A comparison between trajectories from the acquired skill vs. the hand coded policy in simulation.

Summary and Future Work

The Segway RMP and HT present a new and exciting robotics research platform. Based on this platform we have devised a new domain called Segway soccer for investigating human-robot interaction in real-time, adversarial tasks. We have developed the single robot capabilities to control an RMP in an outdoor environment. Specifically, we have developed robust outdoor vision, and extended our prior work with a skill-tactic-play control hierarchy. Our final contribution is a skill acquisition system for rapidly developing new skills through tele-operation.

References

Adams, R., and Bischof, L. 1994. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16:641–647.

Arkin, R. 1998. *Behavior-Based Robotics*. MIT Press.

Asada, M.; Sklar, E.; Kaminka, G. A.; Obst, O.; Polani, D.; Browning, B.; Bonarini, A.; Fujita, M.; Christaller, T.; Takahashi, T.; and Tadokoro, S. 2003. An overview of robocup-2002 fukuoka/busan. *AI Magazine* 24(2):21–40.

Atkeson, C. G.; Moore, A. W.; and Schaal, S. 1997. Locally weighted learning. *Artificial Intelligence Review* 11(1-5):11–73.

Bagnell, J. A.; Kakade, S.; Ng, A.; and Schneider, J. 2003. Policy search by dynamic programming. In *Proceedings of Neural Information Processing Systems, NIPS 16*.

Bar-Shalom, Y. 1990. *Multitarget Multisensor Tracking: Advanced Applications*. Artech House.

Billard, A., and Mataric, M. 2001. Learning human arm movements by imitation: Evaluation of biologicallyinspired connectionist architecture. *Robotics and Autonomous Systems* 37:145–160.

Bowling, M.; Browning, B.; and Veloso, M. 2004, in press. Plays as effective multiagent plans enabling opponent-adaptive play selection. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS'04)*.

Browning, B., and Tryzelaar, E. 2003. Ubersim: A realistic simulation engine for robot soccer. In *Proceedings of Autonomous Agents and Multi-Agent Systems, AAMAS'03*.

Bruce, J., and Veloso, M. 2002. Real-time randomized path planning for robot navigation. In *Proceedings of IROS-2002*. An earlier version of this paper appears in the Proceedings of the RoboCup-2002 Symposium.

Bruce, J.; Balch, T.; and Veloso, M. 2000. Fast and inexpensive color image segmentation for interactive robots. In *Proceedings of IROS-2000*.

Bruce, J.; Bowling, M.; Browning, B.; and Veloso, M. 2003, to appear. Multi-robot team response to a multi-robot opponent team. In *Proceedings of ICRA'03, the 2003 IEEE International Conference on Robotics and Automation*.

Cleveland, W., and Loader, C. 1995. Smoothing by local regression: Principles and methods. Technical report, AT T Bell Laboratories, Murray Hill, NY.

Dias, M. B., and Stentz, A. 2002. Opportunistic optimization for market-based multirobot control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2002*.

Ferraresso, M.; Ferrari, C.; Pagello, E.; Polesel, R.; Rosati, R.; Speranzon, A.; and Zanette, W. 2001. Collaborative emergent actions between real soccer robots. In Stone, P.; Balch, T.; and Kraetzschmar, G., eds., *RoboCup-2000: Robot Soccer World Cup IV*, 297–300. Berlin: Springer Verlag.

Kiat, N.; Ming, Q.; Hock, T.; Yee, Y.; and Yoh, S. 2001. Luckystar ii-team description paper. In Stone, P.; Balch, T.; and Kraetzschmar, G., eds., *RoboCup-2000: Robot Soccer World Cup IV*, 543–546. Berlin: Springer Verlag.

Lenser, S.; Bruce, J.; and Veloso, M. 2001. CMPack: A complete software system for autonomous legged soccer robots. In *Proceedings of the Fifth International Conference on Autonomous Agents*. Best Paper Award in the Software Prototypes Track, Honorary Mention.

Nicolescu, M., and Mataric, M. J. 2003. Natural methods for robot task learning: Instructive demonstration, generalization and practice. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 241–248.

Schaal, S., and Atkeson, C. 1998. Constructive incremental learning from only local information. *Neural Computation* 10(8):2047–2084.