Real-Time Human-Robot Interactive Coaching System with Full-Body Control Interface

Anton Bogdanovych[1], Christopher Stanton[2], Xun Wang[2], Mary-Anne Williams[2]

[1] School of Computing and Mathematics
University of Western Sydney, Australia
A.Bogdanovych@uws.edu.au

[2] Faculty of Engineering and Information Technology
University of Technology Sydney, Australia
{CStanton, XuWang, Mary-Anne}@it.uts.edu.au

Abstract. The ambitious goal being pursued by researchers participating in the RoboCup challenge [8] is to develop a team of autonomous humanoid robots that is capable of winning against a team of human soccer players. An important step in this direction is to actively utilise human coaching to improve the skills of robots at both tactical and strategic levels. In this paper we explore the hypothesis that embedding a human into a robot's body and allowing the robot to learn tactical decisions by imitating the human coach can be more efficient than programming the robot explicitly. To enable this, we have developed a sophisticated HRI system that allows a human to interact with, coach and control an Aldebaran Nao robot through the use of a motion capture suit, portable computing devices (iPhone and iPad), and a head mounted display (which allows the human controller to experience the robot's visual perceptions of the world). This paper describes the HRI-Coaching system we have developed, detailing the underlying technologies and lessons learned from using it to control the robot. The system in its current stages shows high potential for human-robot coaching, but requires further calibration and development to allow a robot to learn by imitating the human coach.

1 Introduction

With autonomous robots becoming increasily prevalent in society, natural and intuitive methods are required to interact, guide and improve robot behaviour. Learning by demonstration, observation and imitation are approaches to learning in which a teacher (or coach) provides examples of the desired robot behaviour. Examples range from a teleoperated robot recording the actions performed by the teacher, to autonomous robots learning to perform actions by watching a human teacher perform a similar action [2]. Likewise, in the realm of virtual agents, imitation learning has been used to teach autonomous agents in gaming environments to perform complex manoeuvres performed by human experts [4].

In this paper we consider how to best teach humanoid robots to play soccer in the RoboCup Standard Platform League (SPL). With people, when one person coaches another, the objective of the coach is to use their relevant expert knowledge and experience to improve the task performance of the person being coached. This knowledge transfer is often verbal, but can be aided by demonstration, pictures, videos, and other forms of communication. However, what is the best way to coach an autonomous robot? In this paper we explore this issue, and present a real-time Human-Robot Interactive Coaching System (HRICS). Our long-term goal is to explore methods of teaching the robot (in real-time) to extend and improve their capabilities without explicit programming. In this paper we focus on one aspect of the system - the teaching of skills by demonstration (imitation learning). To this end, we describe how we use a motion capture suit and portable computing devices (in particular the iPhone) to interact and communicate with the robot.

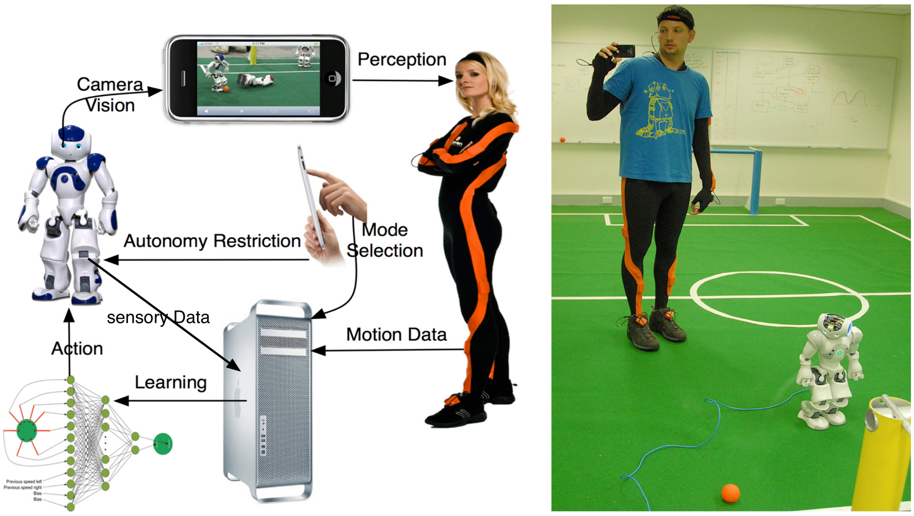
The remainder of the paper is structured as follows. We begin in Section 2 by describing our specific problem domain of robot soccer. In Section 3 we describe our Human-Robot Interactive Coaching System (HRICS). Section 4 outlines the implementation details of how we connected the motion capture suit to the Aldebaran Nao humanoid robot. In Section 5 we present our initial results and reflections regarding using the prototype system to coach soccer skills. We conclude by discussing the potential of the new approach and future work.

2 Problem Domain: Standard Platform League (SPL)

Robot soccer matches in the SPL involve two teams of autonomous Aldebaran Nao [1] robots competing on an indoor field for two 10-minute halves. Human intervention is not allowed during the matches apart from picking up malfunctioning robots or penalising robots that have violated the rules of the game. The Nao robots need to make a wide range of skills autonomously. Perceptual skills include colour recognition, object recognition and the detection of collisions; motor skills include walking and kicking; and there are strategic decisions to be made (for example, positioning on the field). An enormous effort prior to competition is required for teams to develop and calibrate the software to control such skills and behaviours. The prevailing approach for developing such skills is for a software developer to script the behaviour coarsely, but provide a parameterised interface for modifying the behaviour. Most of these parameters will then be calibrated by hand (a very labour intensive process), while a smaller proportion will be calibrated via an unsupervised learning process on the robot [3]. A challenge facing the developers of autonomous robots is how to best specify, develop and improve the robot's skill-set and decision-making capabilities without tediously hand-crafting and hand-tuning behaviours.

3 Human-Robot Coaching System

Human-robot interactive coaching (HRIC) is a new way of approaching the problem of skill learning and development, and performance improvement. In this section we provide a detailed description of our Human-Robot Interactive Coaching System (HRICS) which has been implemented by integrating a combination of



a) b)

Fig. 1. a) System overview and b) System demonstration in the lab.

cutting edge technologies. Figure 1 a) outlines a schematic representation of various components in our system and their interactions; while Figure 1 b) shows a snapshot of the actual system setup that is being tested in our lab.

The focal points of our architecture are the autonomous Nao robot, portable computing devices (iPhone/iPad), a full-body motion capture suit, and data logging for offline processing. The robot is the central part of our architecture it is capable of fully autonomous action, but its autonomy can be restricted by the human coach via the Iphone/Ipad interface. All the data from the robot as well as the user input to the mobile device is stored on the main computer, which processes the data and ensures robot's learning from this data. The perception of the robot is streamed to the mobile device, where it can be annotated for better recognition. The mobile device can also act as a tool for synchronising the perception of the robot and a human coach, which is particularly useful in the imitation mode, where the robot can learn ball searching behaviour, various strategies of approaching the ball or different kicking styles depending on the environment state. While all of the aforementioned components are present in the system and are fully embedded, the main focus of this paper is on integrating its novel element - the full-body motion capture suit. The motion capture suit presents an innovative way of human-robot interaction, where each body part of the human user is involved into controlling the corresponding body part of the robot. The data from the motion capture suit is obtained in real time and is also streamed via WiFi to the main computer, which calculates the necessary transformations to map this data to a robot's motor angles. The technological details of streaming robot vision to a mobile device or collecting robot's sensory data and using supervised machine learning on it [9] are outside of this paper's scope. Next we consider each of the aforementioned system components in detail.

3.1 Aldebaran Nao

The Aldebaran Nao is a humanoid robot with 21 degrees of freedom. The Nao robots are equipped with sensors such as head mounted colour camera, front-facing ultrasonic distance sensors, force sensors in the soles of the feet, bump detectors on the toes of each feet, accelerometers, and gyroscopes.

3.2 iPhone/iPad

We have developed an iPhone/iPad application for interacting with the robot. The application allows the user to interrogate the robot's internal state, stream raw and processed vision in real-time (see Fig. 2), and set operation modes of the robot. The main interface of the application consists of a camera view and a soccer field view which displays raw or processed live images streamed from the robot and the position on the field where the robot believes to be located (Fig. 3). Properly configured robots automatically establish a connection with the portable computer device once they are turned on. An arrowed dot icon is displayed in the soccer field view on the console for every connected robot. The dots on the soccer field correspond to the estimated positions of the robots on the actual field (see Fig. 3).

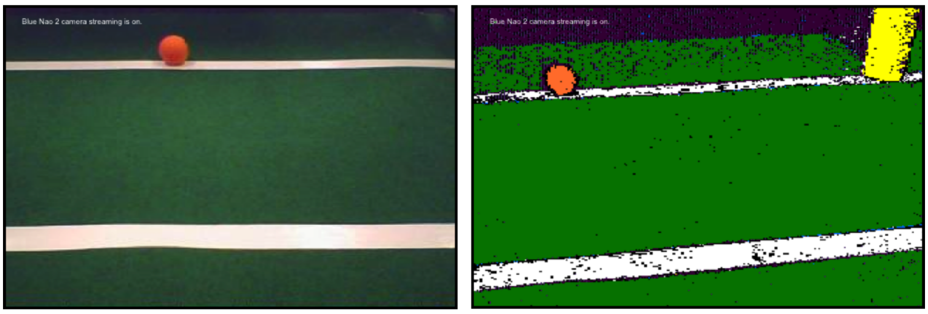


Fig. 2. Raw and processed live images from the robot's camera displayed on an iPhone.

For our current experiments (described in the later sections), we provide the human coach live-feeds of raw vision from the robot's camera. Live images captured from the robot camera are transmitted to an iPhone every 200 ms.

3.3 The Motion Capture Suit

As an interface to control and coach the robot we employ a high precision full-body motion capture suit, Xsens MVN[3]. Only recently motion capture suits

[3] http://xsens.com/en/general/mvn

Fig. 3. The main graphical interface of the iPad interactive coaching application.

similar to Xsens MVN reached the level of precision when they can correctly capture real-time motion of a human body with no significant data errors. This equipment comes in a form of a lycra suit with 17 embedded motion sensors.

The suit is supplied with MVN Studio software that processes raw sensor data and corrects it. It also uses inverse kinematics to cross-verify the data and to estimate the parameters of additional body joints. As the result, MVN studio is capable of sending real-time motion capture data of 23 body segments using the UDP protocol with the frequency of up to 120 motion frames per second. The key elements of the data being transmitted are absolute (X,Y,Z) position of each segment and its absolute (X,Y,Z) rotation. XSENS MVN is capable of real-time motion capture with very high accuracy. During extensive testing it showed a very small margin of error ($0.8°$, $s = 0.6°$ for each of the sensors) [10].

3.4 Data Logging

Commands from the motion capture suit are translated to robot effector commands (the details of this translation are described in Section 4) and sent to the robot. We programmed our robots to log all effector commands, sensor data, and other internal state variables every 10ms[4]. Data is recorded regardless of whether the robot is operating autonomously or being teleoperated via the motion capture suit. If the robot is being controlled via the motion capture suit, this state information is recorded, as are the walk engine commands (forwards, strafe, and rotation) and head position (pitch and yaw) commands chosen by the human controller. This data enables us to analyse (off-line) the decisions made by the human coach in relation to the robot's perceptual state.

[4] On each DCM callback event.

4 Connecting XSENS MVN to a Nao Robot

In our system we directly map each body segment of the human user to the corresponding motor of the robot and adjust those accordingly. The data being transmitted by the suit is measured using the absolute coordinate system of the suit space, where the X axis is aligned with the magnetic north and the origin corresponds to the position at which the suit was turned on. All the rotation data that is being transmitted by the suit comes in the form of quaternions [6].

The robot has a completely different embodiment to a human, so for controlling the robot with a motion capture suit we had to make a number of adjustments. In order to control the movement of its body parts, rather than setting positions and rotations, the robot uses a number of embedded motors, setting the degree of rotation for each of these motors results in the desired movement. To control these motors an SDK allows the programmer to specify the angular rotational position of each motor every 10ms.

When using the motion capture suit for controlling the robot, some of the values we receive for body segments can be directly translated into the appropriate motor rotations on the robot side. For example, there are two motors controlling the head of the robot: pitch motor (headPitch) - responsible for up/down head movement and yaw motor (headYaw) - responsible for side movements. The range of these motors is a bit wider than the corresponding range of the human head motion, but within the range of acceptable human head movement - the angles one must supply to the motors directly map to the Euler angles of the human's head segment.

Fig. 4 provides a graphical explanation for how the data obtained from the motion capture suit is being utilised for controlling the head movement of the robot, its body orientation, as well as forward/backwards/sideways movement of the robot's body. The 3D character shown on the left hand side of the figure corresponds to a reconstructed human model based on the positions and rotations of the body segments received from the motion capture suit. This figure is positioned in the global suit coordinate system, where all coordinates are measured in meters. All the data we receive is in absolute coordinates and angles in relation to this coordinate system.

In order to convert the suit data into the appropriate values for the motor rotations on the robot end, we have to apply the following transformations. First, we must convert the absolute rotation of the head segment into a relative rotation, as the robot operates with angles in the robot space. To do this we have to calculate the relative rotation of the head sensor in relation to the chest segment. The following equation helps to make this translation.

$$Qrot_{relative}(a,b) = \frac{Qrot_b}{Qrot_a} \tag{1}$$

Here $Qrot_{relative}(a,b)$ corresponds to the resulting relative rotation (in the quaternion form) of the head sensor (a) in relation to the chest segment (b). The values of $Qrot_a$ and $Qrot_b$ represent the quaternions defining the absolute rotations of each corresponding body segment.

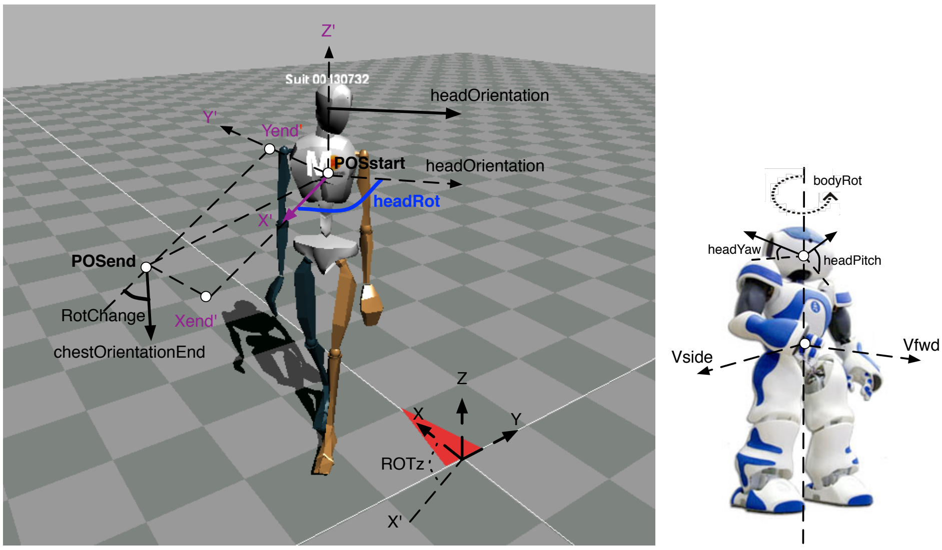


Fig. 4. Translating the MoCap Suit Data into the Robot Motor Angles.

The resulting quaternion rotation (that is shown as headRot in the picture) can be represented as a matrix of the following form:

$$Qrot_{relative}(a,b) = [q_0, q_1, q_2, q_3]^T \tag{2}$$

Once the relative rotation is obtained, we have to transform the quaternion rotation into Euler angles and translate those into correct values for each robot motor. To obtain Euler angles from this matrix - we use the following equation:

$$\begin{vmatrix} rot_X \\ rot_Y \\ rot_Z \end{vmatrix} = \begin{vmatrix} atan2(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ arcsin(2(q_0q_2 - q_3q_1)) \\ atan2(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2))) \end{vmatrix}$$

Here rot_X represents Euler rotation around the X axis, rot_Y is the Euler rotation around the Y axis and rot_Z corresponds to the Euler rotation around the Z axis. In the case of the head, the yaw motor angle (headYaw) is obtained from rot_Z and the pitch (headPitch) - from rot_X.

Controlling the walking movement of the robot is a bit more complicated than rotating the head or arms. The robot has a completely different joint structure to a human in its legs and a significant difference in its centre of gravity. Direct mapping of human joint rotations onto robot motor angles is a difficult task known in the literature as motion retargeting [7]. To avoid obvious mismatches and prevent the robot from falling over we decided not to move every leg sensor individually, but to play prerecorded standard walking moves of the robot in response to the corresponding move being detected as performed by a human.

The robots's SDK provides a function for this purpose, which receives the forward/backward velocity (V_{fwd}), body rotation ($body_{Rot}$) and the side velocity (V_{side}) and results in the desired movement of the robot.

In order to calculate the forward velocity of the robot we first have to compute the forward transformation of the human coach in the direction of the current body orientation. Lets assume that pos_{start} - is the absolute position of the human wearing the motion capture suit in the beginning of forward/backward movement and pos_{end} is the position in the end of this movement.

To obtain the position displacement in the direction of the current body orientation of the human we have to rotate the coordinate system by the angle α, which represents the body rotation of the human (in our case $\alpha = rot_Z$). In order to perform the coordinate rotation we use the following rotation matrix:

$$\Re(\alpha) = \begin{vmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{vmatrix}$$

So, after rotating the coordinates using our rotation matrix, the new coordinates ($X_{start'}, Y_{start'}$) for the point pos_{start} will be:

$$X_{start'} = pos_{start}.X * \cos(\alpha) + pos_{start}.Y * \sin(\alpha)$$
$$Y_{start'} = -pos_{start}.X * \sin(\alpha) + pos_{start}.Y * \cos(\alpha)$$

And the new coordinates ($X_{end'}, Y_{end'}$) for the point pos_{end} will be:

$$X_{end'} = pos_{end}.X * \cos(\alpha) + pos_{end}.Y * \sin(\alpha)$$
$$Y_{end'} = -pos_{end}.X * \sin(\alpha) + pos_{end}.Y * \cos(\alpha)$$

Now we can calculate the forward/backward displacement of the human between those two positions as:

$$\triangle Fwd = X_{end'} - X_{start'} \tag{3}$$

And the sideways displacement of the robot can be obtained as:

$$\triangle Strafe = Y_{end'} - Y_{start'} \tag{4}$$

The displacements $\triangle Fwd$ and $\triangle Strafe$ represent the distances the human body has travelled between two sensor data readings. The data transmission frequency of the suit is manually adjustable and known in advance. Thus to obtain the forward velocity we can use the following equation:

$$V_{fwd} = \frac{\triangle Fwd}{t} \tag{5}$$

And the sideways velocity is obtained as:

$$V_{side} = \frac{\triangle Strafe}{t} \tag{6}$$

In both equations t represents the standard delay between two subsequent suit data transmissions (currently set at 10 ms).

Finally, to make the robot rotate at a desired angle we have to set the value of the corresponding motor. When changing its rotation the robot only deals with relative angles, so it doesn't make any assumptions about its global position and orientation in the world. Similar to obtaining the relative displacements of the sensors we described above, we must compute the change of robot's body rotation in-between two consecutive data measurements of the motion capture suit. To do so, we use the equation (1) and apply it to two measurements of rotations of the chest sensor, so that we obtain the relative rotation ($rotChange = Qrot_{relative}(X', chestOrientationEnd)$) between the chest orientation vector in the first measurement and the chest orientation vector in the last measurement. The *bodyRot* angle that we send to the robot corresponds to the Z Euler angle of the $rotChange$ variable in the suit space (see Fig. 4). Given that $q_0, .., q_2$ are the dimensions of the $rotChange$ quaternion - we can use the equation (2) to calculate the final argument of our function as:

$$body_{Rot} = atan2(2(q_0 q_3 + q_1 q_2), 1 - 2(q_2^2 + q_3^2))) \tag{7}$$

The proportions of the human body and other embodiment characteristics are very different to those of the robot, so no direct mapping can be made between the motion capture data and the data required by the robot. Thus, we had to conduct a series of experiments for obtaining the scaling factors that apply to each of the robot motors.

5 Experimental Evaluation

In order to test the validity of our assumption about the usefulness of the resulting coaching system we conducted a series of experiments where each of the system components was extensively tested[5].

We had a four step plan for evaluating the HRICS. Our first step was calibration of human movements to robot movements via scaling, clipping and the use of minimum movement thresholds. For example, the human controller should not need to run for the robot to walk at top speed; nor should the human fidgeting or other unintentional small movements result in a tiny step from the robot. We calibrated each dimension of the walk engine separately (rotation, strafe and forwards/backwards). Best results were found when small human movements were scaled to larger (proportionally speaking) movements on the robot. For example, for the robot to walk forwards at full speed the human controller was only required to make a small step forward. To remove unintentional human movements from controlling the robot, the human controller was required to stand relatively still. If the robot still moved (due to highly accurate nature of the motion capture suit, even breathing can trigger non-zero values), the minimum value threshold was increased. Head movements were also scaled and offset. As we were using the downwards facing camera of the Nao's two cameras, after

[5] The video recording featuring fragments our experimental evaluation can be seen at: http://www.youtube.com/watch?v=XY4nYpEZr5U

head calibration the robot's "eyes" (they are LEDs very display purposes only) would be facing slightly higher than the human's. After conducting extensive testing we came up with the following adjustments for the robot (see Fig. 4). The $body_{Rot}$ variable had to be magnified by 20. Both V_{fwd} and V_{side} require a magnification by 40. The $headYaw$ didn't require any adjustments, as limiting the range of the robot's head to the range of the human didn't seem to have a significant visual impact. Finally, the $headPitch$ had to be magnified by 1.5 and the values beyond the allowed robot's range had to be clipped.

Next, once the suit was calibrated we attempted to play soccer using the human's vision ("super perception"). This involved the human controller walking about the field, with the robot imitating the human's actions[6]. Scoring a goal was not a simple or straightforward task for the robot. Network lag and the nature of the Aldebaran walk engine meant that the robot would imitate a human step direction approximately 1 second after they executed it. The key reason for this delay was the fact that with our currently chosen approach the robot was not able to interrupt its walk half way through the step cycle, but was only able to change the walking direction after the step cycle was finished. Also, our scaling of human movement to robot movement (despite calibration emphasising that small human movements should be converted to large robot movements), meant the human controller was constantly running out of room to move within the 6m by 4m soccer field. Even more difficult was when the human controller would end up in front of the robot and the ball, and thus they had to somehow move themselves behind the robot, but without the robot moving. Despite these difficulties, the human was able to control the robot in a fluent manner after adjusting his behaviour by moving using smaller steps, and reducing the overall movement velocity so that the robot is able to catch up with him.

After experimenting playing soccer with the human's vision, we head-mounted an iPhone feed of the robot's raw camera feed to the human controller[7]. This allowed the human controller to play soccer in a remote location to the robot, but increased the level of difficulty in playing soccer with the robot, mainly due the small field of view provided by the robot's camera. Having no possibility to see the robot made it difficult to adjust the walking behaviour of the experimenter. As the result, it was often required to perform an additional search for the ball as the experimenter's mental representation of the ball and robot's position on the soccer field didn't match the actual positions. Having the vision, though made it possible to quickly detect reference points and update the mental model.

Lastly, we had planned to try and teach the robot skills by logging human command data, together with robot perceptual data, while controlling the robot suit. However, with our current level of fine control requiring more calibration, an autonomous robot soccer player is a more effective goal-scorer than a robot controlled by the motion-capture suit. However, we are optimistic that with further calibration and refinement, there is a great deal of potential with this

---

[6] This was somewhat confronting visually, as the robot's head would not be looking at the ball. Instead it would be looking in the same direction as the human controller.

[7] In the future, we plan to purchase and use wearable video glasses.

approach. A variety of machine learning algorithms operating on robot sensory data are already embedded into our infrastructure [9], so when we reach the desired level of control precision this part of the experiment will be resumed.

6 Discussion

The development of our Human-Robot Interactive Coaching system is inspired by the real-life human soccer coaching and game plays. It is rooted in the view that robot soccer should adopt successful approaches that have been traditionally applied in the human soccer. Programming every facet of the robot behaviour is time-consuming, inflexible, error-prone and limited. It is crucial to understand that our coaching system does not drive the robot's actions directly as an OCU [5] in a soccer game; it is not purely used for remote control but aims to help the robot learn new skills and strategies. The coaching system monitors the autonomous robot's performance in real-time and gives helping instructions to the robots to improve their game playing skills. Our system is used only in robot training and practice matches, and not in real soccer competitions.

The motion capture suit provides the robots with the possibility to learn rich motion dynamics from the human. These capabilities are difficult to implement using the standard computing hardware. Prior to using the motion capture suit we conducted experiments with remotely controlling a team of robots via keyboard input, but it was too difficult to work with so many parameters simultaneously and was hard to achieve the desired level of precision in setting each of the parameters. Correctly updating the value for each of the 21 motors of the robot by pressing the corresponding keyboard button was a difficult task and remembering the keys that correspond to each of those joints was even more challenging. As the result, the autonomous robots prevailed in our experiments and the human participants reported on the high degree of frustration and confusion with controls. In contrast, the use of motion capture suit allowed for a much better and intuitive interface than a keyboard. In particular, control of the robot's head was fluent and natural (as motor commands for the Nao's head were updated very 100ms, as opposed to control of the legs which was updated on every step-cylce). Due to the number of problems we discussed above it was still not feasible to successfully compete against autonomous robots, but the potential of this interface is very high.

The approach we applied for mimicking human's head rotation was successfully tested on moving the arms of the robot. This functionality, however, had to be disabled to make it possible for the user to utilise his hands for interacting with the mobile device rather than for controlling robot's hand movement.

7 Conclusions and Future Work

We presented the progress made in developing a innovative real-time Human-Robot Interactive Coaching system based on an iPhone/iPad interactive coaching console and Xsens MVN full-body motion capture suit for the autonomous robot Nao's motion and behaviour learning. We have shown that our interactive coaching system is a promising solution to effectively improve the performance of Nao robots in autonomous soccer game play through real-time exchange of rich information between the robots in training and the human coach.

The future work will include further fine-tuning of the human-robot motion correspondence, with the aim of allowing a human controller to become an expert coach via teleoperation. We are planning to utilise machine learning to come up with a mapping function that would create correspondences between motion snapshots recorded by the MoCap suit and the resulting posture of the robot. Once this function is in place, it will be possible to design our own walk engine and avoid the lag problems that we discussed earlier. We will also conduct a series of experiments on teaching various tactical and strategical behaviours to the robot from imitating the human user. In particular, we will focus on learning how to map an existing mental model about the state of the world and sensory perception of the robot to an efficient ball searching strategy by learning it directly from a human. Also we will explore using imitation learning for training the robot various kicking styles and selecting an appropriate kicking style depending on the state of the environment.

References

1. Aldebaran: Nao Robot. http://www.aldebaran-robotics.com
2. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems 57, 469–483 (May 2009), `http://portal.acm.org/citation.cfm?id=1523530.1524008`
3. Chalup, S., Murch, C., Quinlan, M.J.: Machine Learning With AIBO Robots in the Four-Legged League of RoboCup. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37, 297–310 (2007)
4. Gorman, B., Thurau, C., Bauckhage, C., Humphrys, M.: Believability Testing and Bayesian Imitation in Interactive Computer Games. In: Proceedings of the 9th International Conference on the Simulation of Adaptive Behavior (SAB'06). vol. LNAI 4095, pp. 655–666. Springer (2006)
5. Gutierrez, R., Craighead, J.: A native iPhone packbot OCU. In: HRI '09: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction. pp. 193–194. ACM, New York, NY, USA (2009)
6. Kuipers, J.B.: Quaternions and Rotation Sequences. Princeton University Press, Princeton (1999)
7. Nakaoka, S., Nakazawa, A., Yokoi, K., Hirukawa, H., Ikeuchi, K.: Generating whole body motions for a biped humanoid robot from captured human dances. In: IEEE International Conference on Robotics and Automation. pp. 3905–3910 (2003)
8. RoboCup: World Championship and Conference. http://www.robocup.org
9. Stanton, C., Ratanasena, E., Haider, S., Williams, M.A.: Perceiving forces, bumps and touches from proprioceptive expectations. In: RoboCup 2011 Symposium. Springer, Berlin/Heidelberg (2011)
10. Supej, M.: 3D measurements of alpine skiing with an inertial sensor motion capture suit and GNSS RTK system. Journal of Sports Sciences 28(7), 759–769 (2010)