The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Adaptive Motion Control with Visual Feedback for a Humanoid Robot

Heinrich Mellmann* and Yuan Xu*
Department of Computer Science Artificial Intelligence Laboratory
Humboldt-Universität zu Berlin
Unter den Linden 6 10099 Berlin, Germany
{xu,mellmann}@informatik.hu-berlin.de
http://www.naoteamhumboldt.de

***Abstract*— The performance of a soccer robot is highly dependent on its motion ability. The kicking motion is one of the most important motions in a soccer game. However, automatic, full body motion generation for humanoid robots presents a formidable computational challenge. At the current state the most common approaches of implementing this motion are based on key frame technique. Such solutions are inflexible, i.e., in order to adjust the aimed direction of the kick the robot has to walk around the ball. The adjustment costs a lot of time especially if some precise adjustments have to be done, e.g., for a penalty kick. In this paper we present an approach for adaptive control of the motions. We implemented our approach in order to solve the task of kicking the ball on a humanoid robot Nao. The approach was tested both in simulation and on a real robot.**

I. INTRODUCTION

The Robot World Cup (RoboCup) initiative is an attempt to foster artificial intelligence and intelligent robotics research by providing a standard problem where a wide range of technologies can be integrated and examined [1]. In order to actually play a soccer game, a humanoid robot should be able to perform various motions, such as: walking, running, kicking, getting-up, etc. At present, the performance of a soccer robot is highly dependent on its motion ability. Automatic, full body motion generation for humanoid robots presents a formidable computational challenge due to

- the high number of degrees of freedom;
- complex kinematic and dynamic models;
- balance constraints;

Furthermore, in the *dynamic* and *adversarial* environment, such as soccer game, with moving objects where some of them are rational agents playing a game against you, more challenges are raising:

- collision free;
- switching target dynamically;
- cope with unexpected external forces.

In order to succeed in real soccer game, humanoid robots need to perform stable, dynamic and robust motions. However, the humanoid robots of today still do not satisfy the aforementioned demands and their level of dynamically stable mobility is insufficient in the context of the real and uncertain environment.

*Both authors contributed equally to this work.

During the past 30 years, many studies have been conducted on motion of humanoid robots, especially on biped locomotion control, and many methods have been proposed. They can be generally categorized into two groups: The *model based approaches* [2], [3], [4], and the *model free approaches* [5], [6], [7]. In the former, the designer precisely constructs a physical model of the target system (i.e, the robot and environment) and builds a specific controller based on this model. In the latter, it is more important to make use of the intrinsic dynamics of a robot or to associate the sensor information with motions.

The kicking motion is one of the most important motions in a soccer game. At the current state the most common approaches of implementing this motion are key frame based techniques [8], [9] in the RoboCup Standard Platform League. However, such solutions are inflexible, i.e., in order to adjust the aimed direction of the kick the robot has to walk around the ball. Such adjustments cost a lot of time especially if some precise adjustments have to be done, e.g., for a penalty kick. Search techniques, such as Rapidly-exploring Random Trees [10], are applied in full body motion planning of humanoid robot, but only in a static environment. Replanning method [11] based on an off-line computation is proposed to adapt the kicking motion, but only a feasible sub-set of the motion parameters is considered.

In this paper we present an approach for adaptive control of the motions. As an application we implement the adaptive kick on the robot Nao, a humanoid robot used in the RoboCup *Standard Platform League* (SPL).

The paper is organized as follows. In section II, we briefly describe the adaptive motion control with visual feedback for a humanoid robot; the generation of reachable space is described section III; section IV and section V describe the motion planning and stabilization in our application respectively; the experimental results are given in section VI, followed by conclusion and discussion in section VII.

II. ADAPTIVE KICK MOTION CONTROL

In order to implement an adaptive kick for a humanoid robot, which enables robot to kick the ball without a lot of adjustments and adapt according to vision feedback on line, we introduce the adaptive motion control with visual feedback.

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

In our approach we model the motion in Cartesian space, the joint trajectories are generated by inverse kinematic. We have to ensure the adaptivity and to satisfy the conditions line stability at the same time. Thus, it is a obvious idea to describe the motion itself as an optimization problem, e.g., minimize the angle between the foot and the target directory during the kick preparation or maximize the speed of the foot to get stronger kick. To solve a complex optimization problem may be a hard job. To ensure the adaptivity in real time the motion trajectories are not calculated explicitly. Rather, we calculate the next position of the foot in each cycle by local optimization. Since the conditions change according to the sensory input, e.g., seen position of the ball the resulting motion trajectory changes continuously. Of course, it requires the conditions to be defined in a way not to run into an unwanted local minimum or maximum (depending on formulation). In the following we present the basic structure of the kicking motion.

A. Basic Structure of the Kick

The kick is divided in four phases: preparation, retraction, execution and wrap-up phase. In the preparation phase the robot moves the body to one foot and lift the other. In the second phase the robot retract the foot according to the visual input. After retraction is finished, the robot execute the kick. In the last phase the robot put the lifted foot back to the ground and go to the initial position. If the situation changes and kick is impossible anymore, e.g., the ball is too far away, the robot can break up the kick and change directly to the wrap-up phase at any time. The adaptation to the visual input is done in the retraction phase. The stabilization is necessary in all four phases. We will discuss them more detailed in the following sections. In this section we present the basic structure of the kicking motion.

B. Formulation of the Problem

Now we formulate the kicking task geometrically as follows: let $\mathbf{p}_b \in \mathbb{R}^3$ be the *kicking point*, i.e., the point which should be moved by the kicking motion, e.g., the center of mass of the ball. Further, let $\mathbf{v}_b \in \mathbb{R}^3$ with $\|\mathbf{v}_b\| = 1$ be the *intended direction* of the movement after kicking, e.g., direction to the goal. A pair $(\mathbf{p}_b, \mathbf{v}_b)$ consisting a *kicking point* and an *intended direction* is called a *kick request*. The Fig.1 visualizes a *kick request* for ball.

C. Adaptive kick

The whole adaptive kick is proposed as follows: firstly, the target position of kicking foot $\mathbf{p}_f$ is calculated from the *kick request* $(\mathbf{p}_b, \mathbf{v}_b)$; then a motion planning for the kick is made, e.g., the retraction position of kicking foot $\mathbf{p}_r$ is determined. As already mentioned, the actual kick motion is performed in two phases: *retraction* and *execution*: during the retraction the robot moves the kicking foot to $\mathbf{p}_r$ in the preparation, and to $\mathbf{p}_f$ in the execution. All the calculations are on line, i.e., since the kick request changes according to the visual input, the positions of the calculated points $\mathbf{p}_f$ and $\mathbf{p}_r$ changes as well. Therefore the kick can be adapted with vision feedback in real time.

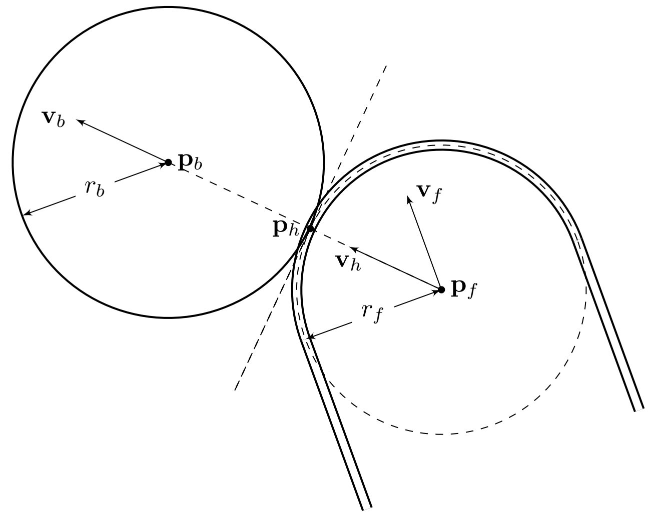

Fig. 1. Formulation of the kicking task. The *kick request* is defined by $(\mathbf{p}_b, \mathbf{v}_b)$, the target of the foot motion is denoted by $(\mathbf{p}_f, \mathbf{v}_f)$, and $\mathbf{p}_h$ is the *hitting spot*. r_b and r_f are the radius of ball and half width of foot respectively. The involved directional vectors(v_b, v_h and v_f) are unit vectors.

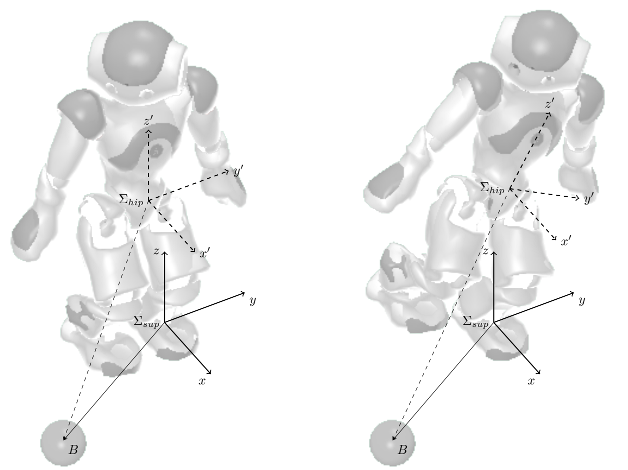

Fig. 2. The position of ball varies in hip coordinates Σ_{hip} during the kicking, but it doesn't vary in support foot coordinates Σ_{sup}.

D. Coordinates

The relation between kicking foot and the ball is crucial in kicking. However, since the robot has to move its body while performing a kicking, it makes problem if the origin of the coordinate system is located in the in the hip as normal case. In our implementation, the support foot is chosen as the origin of coordinates, thus the ball's position doesn't vary during the kicking, see Fig.2.

E. Hitting spot

The *hitting spot* is the collision point between ball and foot while kicking, it determines the movement of ball after kicking. On one hand, in order to reach the target direction of ball movement $\mathbf{v}_b$, the foot movement $\mathbf{v}_f$ should be as

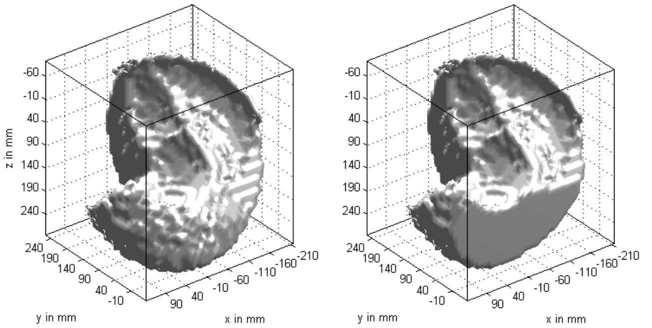

Fig. 3. The *reachable space* of the kicking foot is approximated by a three dimensional grid; (left) the theoretical reachable grid generated in simulation; (right) the grid after experiment on the real robot (flattened manually).

close as possible to $\mathbf{v}_b$. On another hand, in order to achieve powerful kicking, the foot should be retracted as much as possible. But in some cases these two targets can not be reached at the same time. Because the front of Nao's foot is round, the collision between ball and foot can be simplified as collision between two balls, see Fig.1. The *hitting spot* $\mathbf{p}_h$ is calculated to reach the target direction. Thus, we can calculate

$$\mathbf{p}_h = \mathbf{p}_b - \mathbf{v}_b \cdot r_b \tag{1}$$

$$\mathbf{p}_f = \mathbf{p}_h - \mathbf{v}_b \cdot r_f \tag{2}$$

$$\mathbf{v}_h = \mathbf{v}_b \tag{3}$$

Note that $\mathbf{v}_f$ is not determined here, it will be determined by motion planning in the reachable space (See section IV for details).

III. Reachable Space

The *reachable space* of a humanoid robot is defined as the set of points that can be reached by its end effector, with respect to a reference frame of the robot. The *reachable space* is very important in the planning and control of motion and manipulation. Because the adaptive motion control can only adjust the motion trajectory under certain conditions, e.g., in the *reachable space*. HRP-2 uses numerical methods to generate reachable space for arm manipulation[12].

By *reachable space* in the kicking task of a humanoid robot, we mean the space which can be reached by one foot, while the robot stands stably on the another. To generate this space, we define the notation and formulate the basic constraints that a humanoid robot has to satisfy in kicking, including

- the kinematic constraint, e.g.,
  - the limits of joint angles;
  - collision constraint: the end effector shouldn't collide with robot itself.
- balance constraint: the robot should stands stably with one foot.

The *reachable space* of an end effector that rotates and translates in $\mathbb{R}^3$ is a six-dimensional manifold. In order to reduce the number of variables in the space, we decided to represent the space of the reachability by a three dimensional grid. Thereby, we don't consider the rotation of the end effector (e.g., foot), i.e., some of the points in the grid might be reachable only with a special rotation. The main reason for this decision is the simplicity and representational power, the resulting *reachable space* is a subset of $\mathbb{R}^3$. Two experiments are done to generate this *reachable space*.

In the first step we explored the *reachable space* in simulation, i.e., the space contains points which could be reached by the end effector considering only the limitations of the joints (e.g., minimal and maximal reachable angle) and the kinematic constraint.

In the second step we generated the *reachable space* according to the physical limitations in an experiment on the real root. For that, we let the robot move the end effector to every of the reachable points which generated in the first step, and record the reached points at the same time. In this experiment, all the constraints listed above are considered.

Fig. 3 illustrates the *reachable space* of the kicking foot generated by our experiments.

IV. Motion Planning

As already discussed, we can divide the kick in four phases: preparation, retraction, execution and wrap-up phase. In the retraction phase the robot retracts the foot back in order to get load. In the execution phase the robot moves the foot forward as fast as possible towards the hitting point $\mathbf{p}_h$.

Considering this approach there are two questions arising: how to calculate the retracting point and how to calculate the fastest possible kicking trajectory. We will discuss this questions in the following.

In order to analyze these problems we simplify it to a two dimensional case. For that, we assume that the height of the kicking foot constantly equals to the radius of the ball.

A. Retraction Point

To answer the first question from above we consider the requirements for the retraction of the foot. First of all the robot should retract the foot as far as possible from the hitting point $\mathbf{p}_h$ to get the maximal load. This is, of course, a very naive assumption, as the maximal impulse is given by maximal joint velocities and the posture of the robot. Additionally, the retraction point should be chosen in the way that the retracted foot points as much as possible in the requested kicking direction $\mathbf{v}_b$. Of course, the retraction is limited by the reachable space of the foot and also by stabilizing ability of the robot. The problem of stabilization will be discussed later in the section V, for simplicity reasons we assume at this point that the robot can stabilize our motion and focus on the reachability constraints.

In order to express the requirements mentioned above we define the following function

$$g : \mathbb{R}^2 \to \mathbb{R}, \quad \mathbf{p} \mapsto g(\mathbf{p}) := \frac{(\mathbf{p}_h - \mathbf{p})^t \cdot \mathbf{v}_h}{\|\mathbf{p}_h - \mathbf{p}\|}. \tag{4}$$

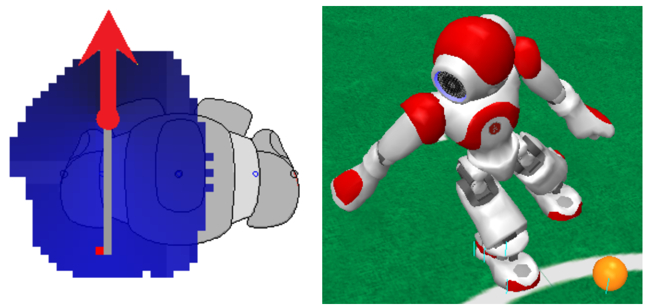

Fig. 4. The retracting point is calculated according to the reachability grid. Left figure illustrates the grid of all reachable positions in the xy-plane (blue) whereas the height is fixed to the radius of the ball. Additionally the requested kick point $\mathbf{p}_f$ and direction $\mathbf{v}_f$ are marked by red arrow. The calculated point of retraction $\mathbf{p}_r$ is marked by a red square (the cell in the reachability grid which maximizes the function f_δ). The figure on the right shows the executed preparing motion according to the calculations shown in the left image.

Obviously, it holds $g(\mathbf{p}) = 1$ if, and only if $(\mathbf{p}_h - \mathbf{p}) = \lambda \mathbf{v}_h$ for a certain $\lambda \in \mathbb{R}$. For easier handling we define the function

$$\hat{g}(\mathbf{p}) := \frac{1 + g(\mathbf{p})}{2}. \tag{5}$$

It holds $\forall \mathbf{p} \in \mathbb{R}^2 : 0 \leq \hat{g}(\mathbf{p}) \leq 1$. We can use this function to satisfy the direction requirement. Now, for a given $\delta \in [0, 1]$ we define the function

$$f_\delta : \mathbb{R}^2 \to \mathbb{R}, \quad \mathbf{p} \mapsto f_\delta(\mathbf{p}) := (1 - \delta) \cdot \|\mathbf{p} - \mathbf{p}_h\| + \delta \cdot \hat{g}(\mathbf{p}) \tag{6}$$

This function combines the conditions for the distance and angle. We can determine the optimal retraction point $\mathbf{p}_r$ as a maximum of f_δ over the reachability grid Ω, i.e.,

$$\mathbf{p}_r = \text{argmax}_{\mathbf{p} \in \Omega}(f_\delta(\mathbf{p})) \tag{7}$$

The parameter δ describes the importance of the angle requirement compared to the distance requirement, i.e., if $\delta = 0$ only the distance is maximized without taking the direction of the kick into account. Note, that δ strongly depends on the size of the reachable space. An optimal δ can be found by experiments, in our tests we used the value $\delta = 1 - 10^{-3}$. Fig. 4 illustrates an example of a calculated retraction point inside the reachability grid.

After finding a point $\mathbf{p}_r$ we can interpret the value $\hat{g}(\mathbf{p}_r)$ as a measure for the precision of the kick, i.e., in the case $\hat{g}(\mathbf{p}_r) = 1$ the direction of the foot movement $\mathbf{v}_f$ corresponds to the desired direction of the kick $\mathbf{v}_h$. This value can be passed to the behavior as a prediction of the kick result. Based on it the behavior could decide whether to finish the kick or to break it up, if its not precise enough.

This approach can be easily extended to the three dimensional case.

B. *Trajectory of the Kick*

After the preparation is done, i.e., the foot reached the retraction point $\mathbf{p}_r$ the robot has to move the foot towards the hitting spot $\mathbf{p}_h$ in order to kick the ball. Usually we want to do it as fast as possible. However, moving the foot along the fastest path may cause problems, e.g., the foot may collide with the ground. In our current implementation we move the foot along the shortest path in the reachability grid, which allows to prevent such collisions. It can be improved by using of the shortest path in the joint space. However, in this case we have to adjust this path according to the reachability grid in order to avoid collisions.

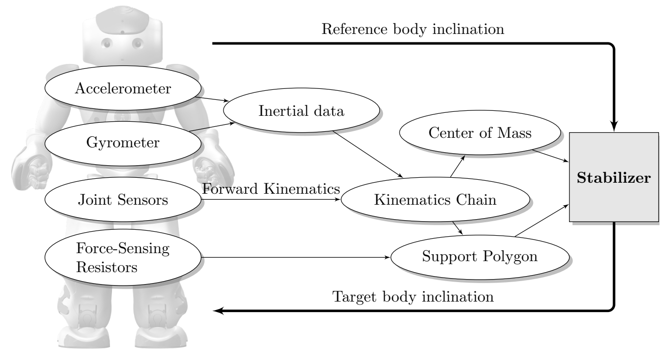

Fig. 5. Closed loop body inclination control: the body inclination is controlled according to center of mass and support polygon which are updated from sensors on line.

V. STABILIZATION

Keeping balance in the single-support phase is one of the major problems. During this phase, the robot is supported only on one foot, so it is more difficult for it to cope with disturbances. Some disturbances, like adjusting kicking foot according to ball, can make the robot lose stability. We introduce feedback control to modify the reference trajectory by using sensor information.

Many stabilization controls have been conducted on walking of humanoid robots, most of them modify the trajectory of foot or hip according to the sensor feedback. B-Human team uses P-controllers to balance the robot according to angles of the torso [9]. ASIMO uses three different posture controls to achieve stable walking: Floor Reaction Control, Target ZMP Control and Foot Planting Location Control [13].

Since the poses of legs and position of the hip is already determined by motion trajectory, the stabilizer can adjust the body inclination to satisfy the static stable criterion, i.e., the center of mass should be in the support polygon. The *Body Inclination Control* is implemented as follows: the center of mass and support polygon is calculated from sensor data, then minimize the difference between center of mass and the center of support polygon by adjusting the body inclination, see Fig.5.

The P control rule is applied as the first trial. The Fig.6 shows that the resulted control system is robust enough, the center of mass tracking errors are damped very fast. The Fig.7 illustrates the center of mass is kept in the center of foot while the robot stabilizes itself during kicking.

VI. EXPERIMENTAL RESULTS

The proposed approach in this paper is implemented in our Nao robot, and some experiments have been done

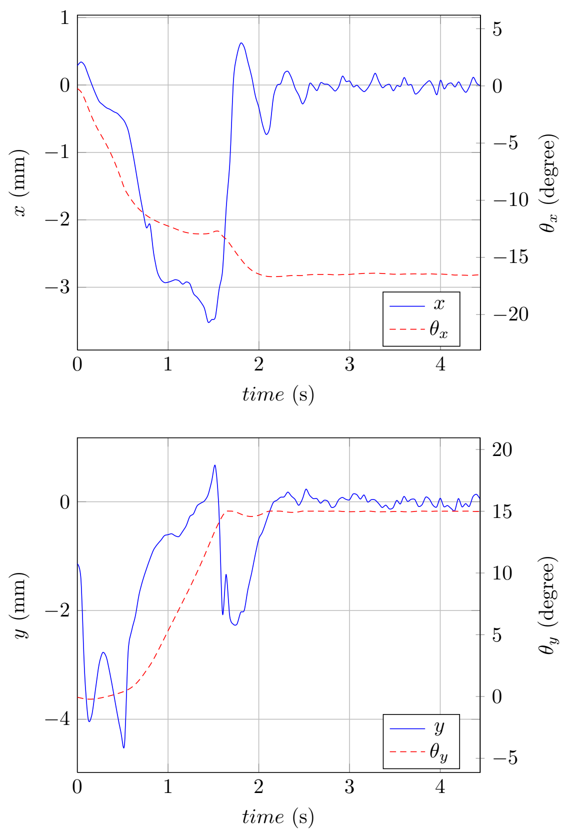

Fig. 6. The tracking errors of center of mass damp with the closed loop body inclination control. The graphs illustrate the behavior of the center of mass while the robot moves from standing by double feet (at time $t = 0$) to the standing on one leg, i.e., prepare for the kick.

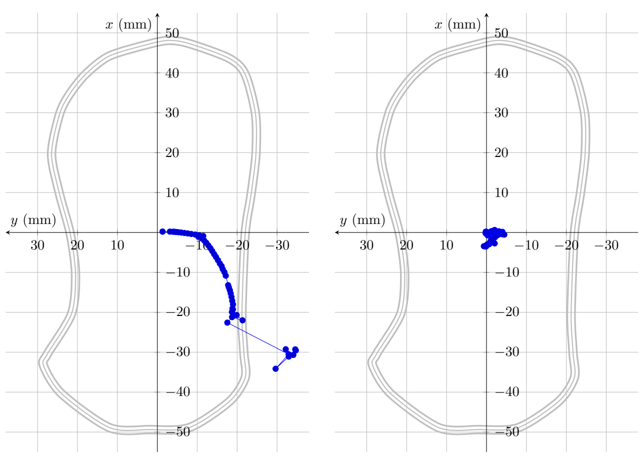

Fig. 7. Robot changes from standing by double feet to the standing on one leg, i.e., prepare for the kick. The left figure shows that center of mass jumps out of support polygon without stabilizer; the center of mass is kept in the center of support polygon with stabilizer in the right figure.

both in simulation and on the real robot. The video showing the experiments performed on the real robot can be found here `http://www.informatik.hu-berlin.de/~naoth/media/video/dynamic_kick.mp4`.

A. Kicking the ball in different positions

The aim of our first experiment is to check the adaptivity performance of the kick. Thus, we let the robot prepare the kick without executing it, i.e., the robot remains in the retraction phase. In this state the robot stands on one foot and adapt the another foot according to the seen ball, which is placed in front of him. Fig.8 illustrates the robot adapting the kick to the changing ball position.

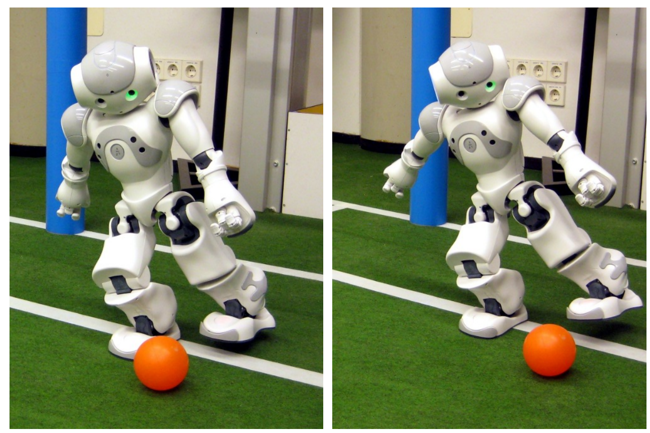

Fig. 8. The robot adapts motion accroding to position of ball, when the intended directions are forward in both situations.

B. Kicking the ball with different intended directions

Analogous the the previous experiment the robot stay in the retraction phase of the motion. Without moving the ball, we request different intended directions of the kick. Fig.9 shows the robot changing the kicking direction from forward (0°) to right (45°).

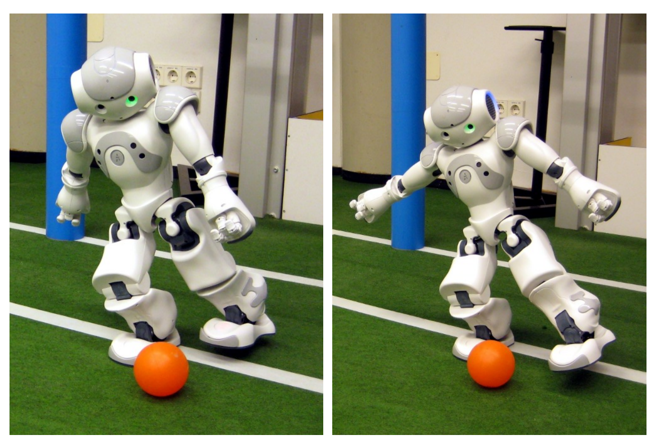

Fig. 9. The robot adapts motion according to the intended direction, the intended direction is forward in the left figure, and it is to the right side in the right figure.

C. Direction accuracy test

In the second experiment we let the robot kick the ball in different directions. After the kick robot observes the moving ball and calculate the error between the intended and the real motion direction of the ball. Fig. 10 illustrates the results of one of these experiments. The bottom figure visualize the trace of the moving ball perceived by the robot after executing the kick. The top graph shows the error in the direction of the kick. It can be seen that stay below 5cm during the first 2m of motion.

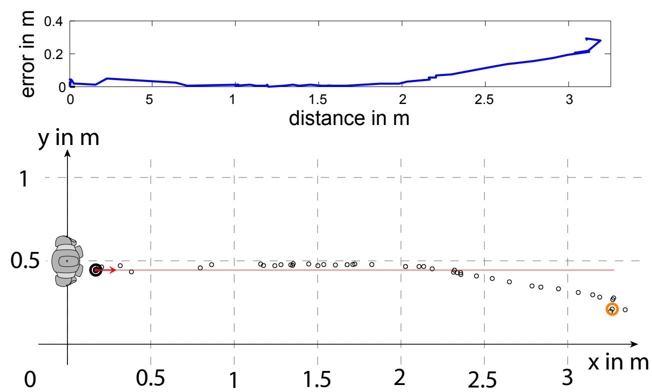

Fig. 10. Results of a kicking experiment: the robot performed a kick strait forward. The bottom figure visualize the positions of the seen ball after the kick. The dark circle represents the initial position of the ball $\mathbf{p}_b$. The arrow depicts the requested direction of the kick $\mathbf{v}_b$. The light circle visualize the resulting position of the ball. The positions of the moving ball are shown by small dots. The top graph visualize the deviation of the moving ball from the intended direction.

The experimental results show that our kicking can adapt for different kicking direction and different ball position. The robot can kick the ball in intended direction without walking around the ball.

VII. Conclusion and Future Work

In this paper, we have presented the adaptive motion control for a humanoid robot, with vision feedback. It enables the robot kick the ball in different positions and to different intended directions. All the calculations are done on line, therefore the kick can be adapted with vision feedback in real time. In the experiments, we studied how the kick motion adapts according to vision feedback. The result shows that our approach is able to accomplish adaptive kick.

In the future, we are interested in learning good strategy and parameters by observation from vision. To achieve this fast, the learning algorithm will be investigated in simulation firstly, and applied to real robot late. Our next step is to include avoiding objects, i.e., avoiding the ball and leg of opponent before kicking, and the dynamic stabilization. Because the support foot slides on the ground sometimes, the friction constraint should be token into account.

The result of the experiments presented in this paper are very promising. Nevertheless, in order to analyze the performance of the kick precisely much more experiments are required.

Acknowledgments

The authors would like to thank the members of the *Nao Team Humboldt*. The research team is also grateful for the comments of the reviewers that help improve the manuscript.

References

[1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: The robot world cup initiative," in *Proceedings of the First International Conference on Autonomous Agents (Agents'97)* (W. L. Johnson and B. Hayes-Roth, eds.), (New York), pp. 340–347, ACM Press, 5–8, 1997.

[2] M. Vukobratovic and B. Borovac, "Zero-moment point- thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[3] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by a simple three-dimensional inverted pendulum model," *Advanced Robotics*, vol. 17, no. 2, pp. 131–147, 2003.

[4] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *ICRA*, pp. 1620–1626, IEEE, 2003.

[5] S. H. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive-dynamic walking robot with two legs and knees," *I. J. Robotic Res.*, vol. 20, no. 7, pp. 607–615, 2001.

[6] S. Grillner, "Neurobiological bases of rhythmic motor acts in vertebrates," *Science*, vol. 228, pp. 143–149, 1985.

[7] M. Ogino, *Embodiment Approaches to Humanoid Behavior*. PhD thesis, Department Of Adaptive Machine Systems, Osaka University, Japan, January 2005.

[8] H.-D. Burkhard, F. Holzhauer, T. Krause, H. Mellmann, C. N. Ritter, O. Welter, and Y. Xu, "Nao-team humboldt 2009," tech. rep., Humboldt Universität zu Berlin, 2009.

[9] T. Röfer, T. Laue, J. Müller, O. Bösche, A. Burchardt, E. Damrose, K. Gillmann, C. Graf, T. J. de Haas, A. Härtl, A. Rieskamp, A. Schreck, I. Sieverdingbeck, and J.-H. Worch, "B-human team report and code release 2009," 2009. Only available online: http://www.b-human.de/download.php?file=coderelease09_doc.

[10] J. J. K. Jr., S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," *Auton. Robots*, vol. 12, no. 1, pp. 105–118, 2002.

[11] S. Lengagne, N. Ramdani, and P. Fraisse, "Planning and fast re-planning of safe motions for humanoid robots : Application to a kicking motion," in *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (St. Louis, USA), October 2009.

[12] Y. Guan, K. Yokoi, and X. Zhang, "Numerical methods for reachable space generation of humanoid robots," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 935–950, 2008.

[13] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *ICRA*, pp. 1321–1326, 1998.