Adaptive Motion Control: Dynamic Kick for a Humanoid Robot

Yuan Xu and Heinrich Mellmann [*]

Institut für Informatik, LFG Künstliche Intelligenz
Humboldt-Universität zu Berlin, Germany
`{xu,mellmann}@informatik.hu-berlin.de`
`http://www.naoteamhumboldt.de`

Abstract. Automatic, full body motion generation for humanoid robots presents a formidable computational challenge. The kicking motion is one of the most important motions in a soccer game. However, at the current state the most common approaches of implementing this motion are based on key frame technique. Such solutions are inflexible and cost a lot of time to adjust robot's position. 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.

1 Introduction

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

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. 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 [1–5]. Search techniques, such as Rapidly-exploring Random Trees [6], are applied in the full body motion planning of humanoid robot, but only in a static environment. 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 [7, 8]. However, such solutions are inflexible, i.e., in order

---

[*] Both authors contributed equally to this work. The research team is grateful for the comments of the reviewers that help improve the manuscript.

to adjust the aimed direction of the kick the robot has to walk around the ball. Such adjustments cost a lot of time especially when some precise adjustments have to be done, e.g., for a penalty shoot. Re-planning method [9] based on an off-line computation is proposed to adapt the kicking motion, but a feasible sub-set of the motion parameters is considered only. 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 2, we describe our approach to the adaptive motion control for a humanoid robot; the experimental results are given in section 3, followed by conclusion and discussion in section 4.

2 Adaptive Kick Motion Control

In order to enable the robot to perform an adaptive motion we have to consider the following three aspects: reachable space, motion planning and stabilization.

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 like stability at the same time. Thus, it is an 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 a strong kick. Solving such complex optimization problem may be a difficult job. In order 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 unexpected local minimum or maximum (depending on formulation). In the following we present the basic structure of the kicking motion.

The kick is divided into 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 goes 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.

Now we formulate the kicking task geometrically as follows: The input of the algorithm, i.e., the *kick request* is given by a pair $(\mathbf{p}_b, \mathbf{v}_b) \in \mathbb{R}^3 \times \{\mathbf{v} \in \mathbb{R}^3 : \|\mathbf{v}_b\| = 1\}$, where $\mathbf{p}_b$ is the point which should be moved (e.g., the center of mass of the ball) and $\mathbf{v}_b$ denotes the desired movement direction of the ball (e.g., direction to the goal). Because the front of Nao's foot is round, the collision

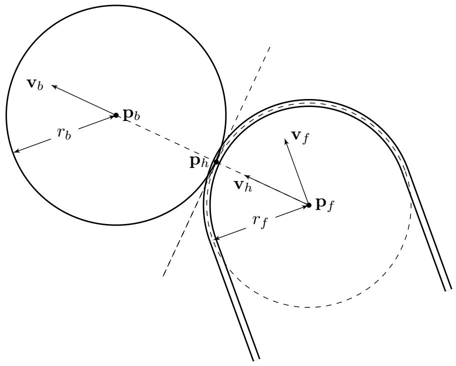

Fig. 1. The *kick request* is defined by $(\mathbf{p}_b, \mathbf{v}_b)$, the target of the foot motion is denoted by $(\mathbf{p}_f, \mathbf{v}_f)$, and $\mathbf{p}_h$ is the *hitting spot*. r_b and r_f are the radius of ball and half width of foot respectively. The involved directional vectors (v_b, v_h and v_f) are unit vectors. We can calculate $\mathbf{p}_h = \mathbf{p}_b - \mathbf{v}_b \cdot r_b$, $\mathbf{p}_f = \mathbf{p}_h - \mathbf{v}_b \cdot r_f$, $\mathbf{v}_h = \mathbf{v}_b$. Direction $\mathbf{v}_f$ results from the motion planning.

between ball and foot can be simplified as collision between two balls, as shown in Fig.1. Thus, the motion direction after the collision is defined by the *hitting spot* $\mathbf{p}_h$, i.e. the collision point between ball and foot. Consequently, the actual task is to move the foot in a trajectory which crosses the point $\mathbf{p}_f$, where the foot collides with the ball. Note that motion direction $\mathbf{v}_f$ results from the motion trajectory and has only influence of the strength of the kick.

2.1 Reachable Space

The *reachable space* of a 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. HRP-2 uses numerical methods to generate reachable space for arm manipulation[10]. 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. The *reachable space* is defined by some basic constraints that a humanoid robot has to satisfy during the kick, including the kinematic constraint (e.g., the limits of joint angles; and the collision constraint.) and the 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 $\mathbb{R}^3$. We generated the *reachable space* according to the physical limitations in an experiment on the real robot. For that, we let the robot move the end effector to every of the reachable points which generated by kinematic constraint, and record the reached points at the same time. In this experiment, all the constraints listed above are considered. Fig. 2 (left) illustrates the *reachable space* of the kicking foot generated by our experiments.

2.2 Motion Planning

As already discussed, we can divide the kick into four phases: preparation, retraction, execution and wrap-up phase. Considering this approach there are two questions arising: how to calculate the retracting point and how to calculate the fastest possible kicking trajectory.

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.

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 2.3, 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

$$g : \mathbb{R}^2 \to \mathbb{R}, \quad \mathbf{p} \mapsto g(\mathbf{p}) := \frac{(\mathbf{p}_h - \mathbf{p})^t \cdot \mathbf{v}_h}{\|\mathbf{p}_h - \mathbf{p}\|}. \tag{1}$$

Obviously, it holds $g(\mathbf{p}) = 1$ if, and only if $(\mathbf{p}_h - \mathbf{p}) = \lambda \mathbf{v}_h$ for a certain $\lambda \in \mathbb{R}$. We can use this function to satisfy the direction requirement. Now, for a given $\delta \in [0, 1]$ we define the function

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

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.,

$$\mathbf{p}_r = \operatorname{argmax}_{\mathbf{p} \in \Omega}(f_\delta(\mathbf{p})) \tag{3}$$

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, we

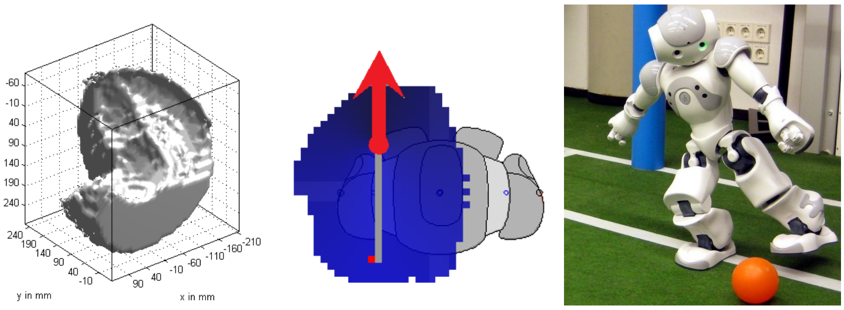

Fig. 2. The *reachable space* of the kicking foot is approximated by a three dimensional grid is shown on the left. The retracting point is calculated according to the reachability grid. Middle figure illustrates the grid of all reachable positions in the xy-plane (blue) whereas the height is fixed to 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.

used the value $\delta = 1 - 10^{-3}$ in our tests. Fig. 2 illustrates an example of a calculated retraction point inside the reachability grid.

After finding a point $\mathbf{p}_r$ we can interpret the value $g(\mathbf{p}_r)$ as a measure for the precision of the kick, i.e., in the case $g(\mathbf{p}_r) = 1$ 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 it is not precise enough.

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

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 $\mathbf{p}_h$ 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.

2.3 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

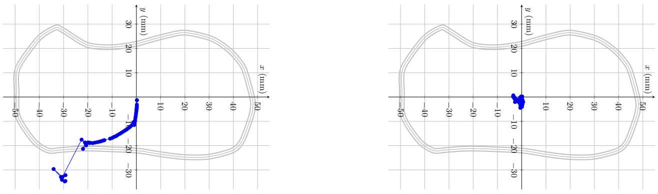

Fig. 3. Robot changes from standing by double feet to the standing on one leg, i.e., prepare for the kick. The left figure shows that center of mass jumps out of support polygon without stabilizer; the center of mass is kept in the center of support polygon with stabilizer in the right figure.

to ball, can make the robot lose stability. We introduce a feedback control to modify the reference trajectory according to the sensor information.

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: in the first step the center of mass and support polygon are calculated from sensor data, in the second step the body inclination is adjusted to minimize the difference between center of mass and the center of support polygon. The P control rule is applied as the first trial. The Fig.3 illustrates the center of mass is kept in the center of foot while the robot stabilizes itself during the kicking.

3 Experimental Results

The approach proposed in this paper was implemented and tested on a real Nao robot and in simulation. In this section we present the results of experiments we performed to evaluate the kick motion. A 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`.

In order to test the adaptivity performance of the proposed kick motion we performed two different experiments: in the first experiment we let the robot kick from the same position but to different directions between forward (0°) to the left (90°). The direction is changed in 10° steps. For each direction we executed 5 kicks. In the second experiment we let the robot kick straight forward, thereby the position of the ball is changed randomly. In this experiment we executed 100 kicks.

The Fig. 4 and Fig. 5 illustrate the results of the first experiment. Here it can be observed, that the resulting direction of the kick has a general negative offset. Another observation is the "hole"for the kick directions around 20°. Both can be explained with the insufficiency of the analytical assumptions for the kick geometry made, e.g., the foot has not a precise form of a circle and the kick is not fully elastic. The Fig. 5 shows the results of the second experiment. Here we

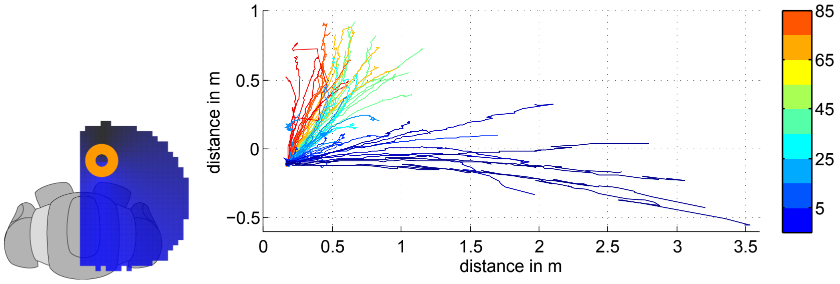

Fig. 4. The kick is performed with the same starting position of the ball but in different directions: (left) position of the ball and the reachability grid; (right) observed trajectories of the ball for each kick direction (respective colors);

can see that the direction of the kick doesn't vary a lot for changing position of the foot.

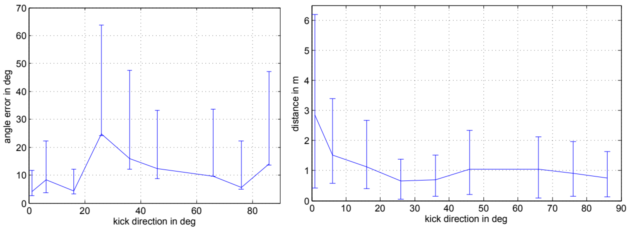

Fig. 5. (left) the angle error of the resulting position of the ball; (right) the maximal kick distance of the ball for each kick direction;

It can be said that the results of the experiments are very promising. But, they also show that there is a need of more sophisticated algorithms (e.g., neural networks) for the adaptation of the foot to ensure precise kicks to any direction.

4 Conclusion and Future Work

In this paper, we have presented the adaptive motion control for a humanoid robot. It enables the robot to kick the ball in different positions and to different intended directions. All calculations are done on line, therefore the kick can be adapted with vision feedback in real time. In our experiments we could show that the presented 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

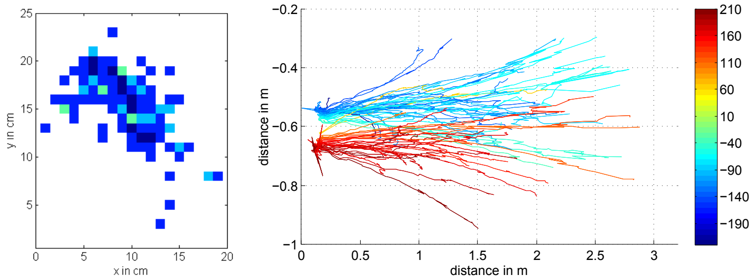

Fig. 6. The kick is performed with the same direction but from random positions: (left) positions of the ball. The color indicates the number of kicks from the according position (the greener, the more); (right) observed trajectories of the ball for each kick position (respective colors);

before kicking, and the dynamic stabilization during kicking execution is also very important.

References

1. Vukobratovic, M., Borovac, B.: Zero-moment point- thirty five years of its life. International Journal of Humanoid Robotics **1**(1) (2004) 157–173
2. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by a simple three-dimensional inverted pendulum model. Advanced Robotics **17**(2) (2003) 131–147
3. Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., Hirukawa, H.: Biped walking pattern generation by using preview control of zero-moment point. In: ICRA, IEEE (2003) 1620–1626
4. Collins, S.H., Wisse, M., Ruina, A.: A three-dimensional passive-dynamic walking robot with two legs and knees. I. J. Robotic Res. **20**(7) (2001) 607–615
5. Grillner, S.: Neurobiological bases of rhythmic motor acts in vertebrates. Science **228** (1985) 143–149
6. Jr., J.J.K., Kagami, S., Nishiwaki, K., Inaba, M., Inoue, H.: Dynamically-stable motion planning for humanoid robots. Auton. Robots **12**(1) (2002) 105–118
7. Burkhard, H.D., Holzhauer, F., Krause, T., Mellmann, H., Ritter, C.N., Welter, O., Xu, Y.: Nao-team humboldt 2009. Technical report, Humboldt Universität zu Berlin (2009)
8. Röfer, T., Laue, T., Müller, J., Bösche, O., Burchardt, A., Damrose, E., Gillmann, K., Graf, C., de Haas, T.J., Härtl, A., Rieskamp, A., Schreck, A., Sieverdingbeck, I., Worch, J.H.: B-human team report and code release 2009 (2009) Only available online: http://www.b-human.de/download.php?file=coderelease09_doc.
9. Lengagne, S., Ramdani, N., Fraisse, P.: 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)
10. Guan, Y., Yokoi, K., Zhang, X.: Numerical methods for reachable space generation of humanoid robots. The International Journal of Robotics Research **27**(8) (2008) 935–950