2016 IEEE-RAS 16th International Conference on
Humanoid Robots (Humanoids)
Cancun, Mexico, Nov 15-17, 2016

Achievement of Dynamic Tennis Swing Motion by Offline Motion Planning and Online Trajectory Modification Based on Optimization with a Humanoid Robot

Ryo Terasawa[1], Shintaro Noda[1], Kunio Kojima[1], Ryo Koyama[1], Fumihito Sugai[1], Shunichi Nozawa[1], Yohei Kakiuchi[1], Kei Okada[1] and Masayuki Inaba[1]

***Abstract*— In order for a humanoid robot to achieve higher physical performance, it is important to generate dynamic whole-body motion under the constraints of physical limitations and dynamic balance. In this paper, we propose the method for generating dynamic whole-body motion such as sports motion based on optimization techniques. Taking a tennis forehand swing as an example of dynamic motion, we aim to increase the swing speed. The proposed methods are composed of the following two methods: 1) the offline swing optimization and 2) the online swing modification. In the offline swing optimization, we use non-linear optimization techniques to maximize the swing speed and satisfy the constraints. We generate all joint trajectories by optimizing control points of uniform B-splines. In the online swing modification, we modify a part of the optimized trajectories online considering joint velocity limits, since the predicted ball trajectory might change before a robot hits the ball. These methods are validated through the following experiments. First, we carry out the experiment in which the actual robot JAXON executes the optimal swing motion. We confirm that the method of the offline swing optimization generates the feasible motion which reaches up to 14.6m/s. We also apply the online swing modification to the optimized motion in the simulation. Then we evaluate how accurately we have to predict the ball trajectory.**

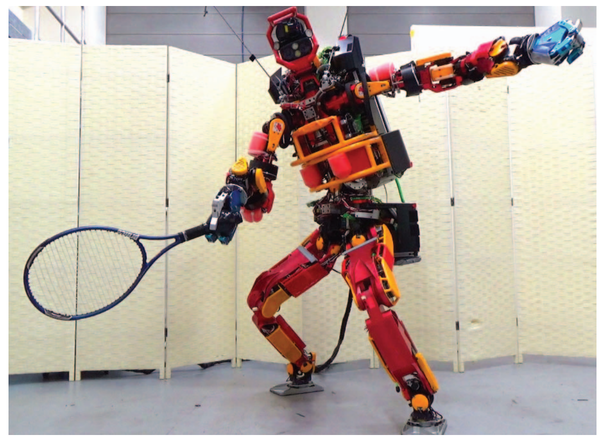


Fig. 1. Humanoid robot JAXON [1] which swings a tennis racket

I. Introduction

The research of humanoid robots has been an active area in recent years. Its human-like shape makes it possible to work directly in the same human environment. Therefore it is expected not simply to walk stably but to perform various tasks such as daily life support or disaster response. Nevertheless, it is still challenging for a humanoid robot to achieve human-level performance with respect to dynamic motion, especially like sports motion.

In this paper, taking a tennis forehand swing as an example of dynamic motion, we aim to increase the swing speed under the constraints of physical limitations and dynamic balance. The proposed methods consist of the two methods: the offline swing optimization and the online swing modification. First, we present the method of the offline swing optimization for generating dynamic whole-body motion. By means of solving a non-linear optimization problem, we generate the optimal swing motion which takes full advantage of humanoid's physical performance. Furthermore, we propose the method of the online swing modification to modify a part of the optimal trajectories. This method is applied according to a change of the reference hitting pose, since the predicted ball trajectory might change before a robot hits the ball. These two methods are verified through the experiments. First, we conduct the experiment in which an actual robot carry out the optimal swing motion generated by the offline swing optimization. We confirm that the planned motion is feasible without falling down or self-collision. In addition, we apply the method of the online swing modification to the optimal swing motion in the simulation. Then we evaluated the method as an index of the required accuracy of the predicted ball position.

[1] R. Terasawa, S. Noda, K. Kojima, R. Koyama, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada and M. Inaba are with Graduate School of Information Science and Technology, The University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan terasawa@jsk.t.u-tokyo.ac.jp

II. Humanoid Swing Motion

A. Related Work

Several studies have reported sports motion generation with a humanoid robot. The representative example is a humanoid robot playing table tennis. For example, the method for a humanoid's keeping stability while playing table tennis was presented in [2]. In addition, the whole system was designed including the real-time vision system, prediction for ball trajectory, and motion trajectory planning for the ping-pong racket in [3]. In these researches, a humanoid robot can play table tennis and rally with a human player successfully.

In terms of a humanoid's swing motion generation, tennis forehand swing and baseball batting motion have been presented so far. The former research introduced the method for encoding desired trajectories through the imitation of human movements, although the balance was not taken into account [4]. The latter showed that a humanoid robot whose swinging forms were taught directly or captured from humans could swing fast with bipedal balancing [5].

However, the above-mentioned motion is not necessarily optimal swing motion. Although the authors analyzed the golf swing motion of elite golfers whose swing is assumed to be optimal, they used not a humanoid robot but a musculoskeletal model to create the dynamic simulation [6]. In this paper, we generate dynamic whole-body swing motion of a tennis forehand swing with a humanoid robot. In order for a humanoid robot to achieve a higher level of physical performance, we optimize all joint trajectories simultaneously for dynamic swing motion.

B. Overview of Offline Swing Optimization and Online Swing Modification for Dynamic Swing Motion Generation

This paper presents the methods of the offline swing optimization and the online swing modification. Fig.2 shows the overview of our system.

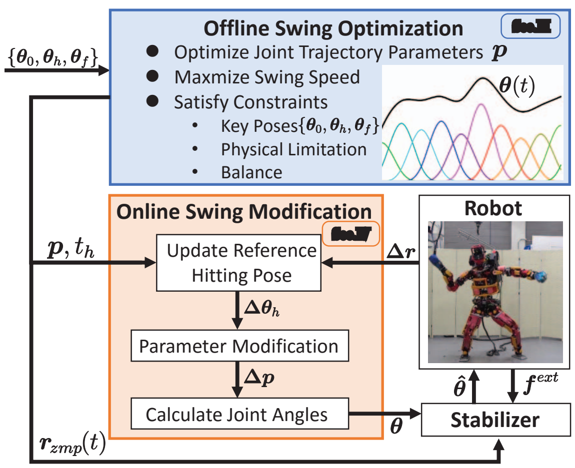


Fig. 2. Overview of our system

The method for optimizing swing motion offline is as follows: we use the uniform B-splines for constructing all joint trajectories. We formulate the non-linear optimization problem which is composed of the objective function for increasing the swing speed and the necessary constraints. Then we optimize the spline parameters (control points) by solving the optimization problem.

The method for modifying swing motion online is as follows: first, the reference hitting pose is updated according to a change of the predicted ball position. Then, the optimal swing motion parameters are successively modified by adding the modification amount. The modification amount is computed by solving the quadratic programming problem which consists of the constraints on the updated reference hitting pose and the other required constraints.

III. Offline Swing Optimization

In order to generate dynamic whole-body swing motion, the method of the offline swing optimization uses all degrees of freedom (DoF) including 1 DoF of each joint and 6 DoF of the root link. Specifically, the proposed method optimizes both the swing speed and the hitting time t_h by means of the non-linear optimization. It also satisfies the joint angles of the initial pose $\boldsymbol{\theta}_0$ at the start time $t = 0$, the final pose $\boldsymbol{\theta}_f$ at the given finish time $t = t_f$, and the hitting pose $\boldsymbol{\theta}_h$ at the hitting time $t = t_h$. Moreover, it should satisfy the constraints of the physical limitations and the balancing of the motion. Accordingly, the optimization problem is formulated as follows:

$$\min_{\boldsymbol{\theta}(t), t_h} f(t, \boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(n)}) \tag{1}$$

$$\text{subject to} \quad \boldsymbol{g}(t, \boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(n)}) = \boldsymbol{0} \tag{2}$$

$$\boldsymbol{h}(t, \boldsymbol{\theta}^{(0)}, \boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(n)}) \leq \boldsymbol{0} \tag{3}$$

where $\boldsymbol{\theta}^{(n)}$ is the nth time derivative of $\boldsymbol{\theta}$.

A. Properties of the Uniform B-splines

In the field of the robot motion planning, the generation of the desired motion is equivalent to finding the joint trajectories. The optimization techniques are widely utilized to find the optimal joint trajectory parameters. In this paper, we use n-order uniform B-splines as a function of joint trajectories. This means that a certain joint j's trajectory function $\theta_j(t)$ is written as

$$\theta_j(t) = \sum_{i=0}^{m-1} p_{j,i} b_{i,n}(t) \tag{4}$$

where m is the number of control points $p_i (i = 0, \cdots, m-1)$, and $b_{i,n}(t)$ is the basis function of order n computed by the Cox-de Boor recursion [7]. The properties of the uniform B-splines are [8]:

1) Convex hull property: The B-spline curves are contained in the convex hull of its control points, that is

$$p_{\min} \leq p_i \leq p_{\max} \Rightarrow p_{\min} \leq \theta(t) \leq p_{\max} \tag{5}$$

2) Local property: The value of the trajectory function at a certain time $\hat{t}$ depends only on the basis function around $\hat{t}$, which means

$$\theta(\hat{t}) = \sum_{i=0}^{m-1} p_i b_{i,n}(\hat{t}) = \sum_{i=i^*} p_i b_{i,n}(\hat{t}) \tag{6}$$

where i^* is the indices such that $t_i - \frac{n}{2}h \leq \hat{t} \leq t_i + \frac{n}{2}h$ for any knots t_i. h is the interval between adjacent knots.

3) Differentiability: The n-order uniform B-splines is C^{n-1} function. The time derivative of $\theta(t)$ is computed as follows:

$$\dot{\theta}(t) = \sum_{i=0}^{m-2} p_i \dot{b}_{i,n}(t) = \sum_{i=0}^{m-2} q_i b_{i,n-1}(t) \tag{7}$$

$$q_i = \frac{1}{h}(p_{i+1} - p_i) \tag{8}$$

Each joint trajectory is constructed by the uniform B-splines. The control point vector $\boldsymbol{p} = [\boldsymbol{p}_0^T \cdots \boldsymbol{p}_j^T \cdots \boldsymbol{p}_{K-1}^T]^T \in \mathbb{R}^{m \times K}$ is computed by solving the optimization problem. K is the sum of each joint DoF and 6 DoF of the root link.

B. Objective Function

In order to maximize the swing speed, the objective function contains a racket linear velocity when the joint angles of the robot correspond to the hitting pose $\boldsymbol{\theta}_h$. The racket linear velocity means the one in a normal direction of the racket surface. The hitting time t_h is also optimized simultaneously.

However, only this condition of the swing speed might generate improper motion which overloads motors of the robot. Therefore, we use the minimum-jerk model for human movements in order to generate smoother motion [9].

Consequently, we define the objective function $\min_{\boldsymbol{p},t_h} f(t, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}, \dddot{\boldsymbol{\theta}})$ of Eq. 1 as follows:

$$\min_{\boldsymbol{p},t_h} \left(-w\boldsymbol{n}^T \boldsymbol{J}(\boldsymbol{\theta}_h)\dot{\boldsymbol{\theta}}(t_h) + (1-w)\int_0^{t_f} ||\dddot{\boldsymbol{\theta}}(t)||^2 dt \right) \quad (9)$$

where $w > 0$ is a variable for weighting, $\boldsymbol{n}$ is a unit normal vector to the racket surface, and $\boldsymbol{J}(\boldsymbol{\theta}_h)$ is the Jacobian for the racket (sweet spot) when the joint angles are $\boldsymbol{\theta}_h$.

C. Equality Constraints

The equality constraints include satisfying joint angles $\boldsymbol{\theta}_0$ at the start time $t = 0$, $\boldsymbol{\theta}_h$ at the hitting time $t = t_h$ and $\boldsymbol{\theta}_f$ at the finish time $t = t_f$. Furthermore, it is desirable that joint velocities and joint accelerations at the start and finish time of the motion equals to zero in order not to damage hardware. Thus, the equality constraints on joint angles, joint velocities, and joint accelerations are written as

$$\begin{cases} \boldsymbol{\theta}(0) = \boldsymbol{\theta}_0, \dot{\boldsymbol{\theta}}(0) = \ddot{\boldsymbol{\theta}}(0) = \boldsymbol{0} \\ \boldsymbol{\theta}(t_h) = \boldsymbol{\theta}_h \\ \boldsymbol{\theta}(t_f) = \boldsymbol{\theta}_f, \dot{\boldsymbol{\theta}}(t_f) = \ddot{\boldsymbol{\theta}}(t_f) = \boldsymbol{0} \end{cases} \quad (10)$$

Additionally, we generate the swing motion of a humanoid robot with its feet positions fixed, which means

$$\begin{cases} \boldsymbol{r}_L(t) = \boldsymbol{r}_L \ (Const.) \\ \boldsymbol{r}_R(t) = \boldsymbol{r}_R \ (Const.) \end{cases} \quad (0 \le t \le t_f) \quad (11)$$

where $\boldsymbol{r}_L(t)$, $\boldsymbol{r}_R(t)$ is the left or right foot position at time t respectively. The constant values $\boldsymbol{r}_L$, $\boldsymbol{r}_R$ are given.

In summary, the equality constraints $\boldsymbol{g}(t, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) = \boldsymbol{0}$ of Eq. 2 include Eq. 10 and Eq. 11.

D. Inequality Constraints

The inequality constraints $\boldsymbol{h}(t, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) \le \boldsymbol{0}$ of Eq. 3 are composed of the physical limitations and the balance of the robot. The details are described in the following.

1) Physical limitations: The physical limitations contain the constraints on the range of the joint angles or joint velocities, and self-collision avoidance. The former constraints are written as Eq. 12. We use the convex-hull property of B-splines described in Sec. III-A.

$$\theta_{j\min} \le p_{j,i} \le \theta_{j\max}, \dot{\theta}_{j\min} \le q_{j,i} \le \dot{\theta}_{j\max} \quad (12)$$

$q_{j,i}$ represents the control points of the joint velocity trajectories computed with Eq. 8. $\theta_{j\max}$, $\theta_{j\min}$, $\dot{\theta}_{j\max}$ and $\dot{\theta}_{j\min}$ are the upper or lower limit of the joint angle or joint velocity of joint j respectively.

The latter constraints are simply written as

$$d(\mathcal{L}_i, \mathcal{L}_j) \ge \epsilon_c \quad (0 \le t \le t_f,\ i \ne j,\ j \ne i-1) \quad (13)$$

where $\mathcal{L}_i$ means the ith link of the robot and ϵ_c is a security margin [8]. In this study, for simplicity, we calculate the distance between $\mathcal{L}_i$ and $\mathcal{L}_j$, using the representative point of each link. The checked pairs are determined beforehand: the forearm of the racket arm and the chest, torso, or upper arm of the non-racket arm.

2) Balance: When a humanoid robot performs the swing motion, its balance needs to be maintained. Thus we set the condition that the ZMP trajectory calculated from the motion lies inside the support polygon [10]. We take the robot dynamics into account in calculating the ZMP trajectory.

Let $\mathcal{S}(\boldsymbol{r}_L, \boldsymbol{r}_R)$ be the region of the support polygon determined by the foot position $\boldsymbol{r}_L$ and $\boldsymbol{r}_R$, $\boldsymbol{r}_{zmp}(t)$ be the position of the ZMP at time t, and $d\left(\mathcal{S}, \boldsymbol{r}_{zmp}\right)$ be the minimum distance between the ZMP and each side of a support polygon. Then the balance constraints are represented as follows:

$$\delta \cdot \left(d\left(\mathcal{S}, \boldsymbol{r}_{zmp}\right) - \epsilon_b\right) \ge 0 \quad (0 \le t \le t_f) \quad (14)$$

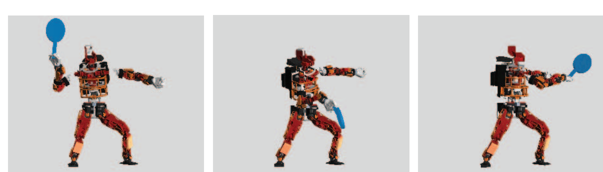
where ϵ_b is a margin and δ is as below.

$$\delta = \begin{cases} 1 & \text{if } \ \boldsymbol{r}_{zmp}(t) \in \mathcal{S}(\boldsymbol{r}_L, \boldsymbol{r}_R) \\ -1 & \text{otherwise} \end{cases} \quad (15)$$

In this paper, Eq. 11, Eq. 13 and Eq. 14 are discretized by a discrete time respectively whose interval is short enough.

E. Motion Generation

In this study, we generate the optimal tennis swing motion for the life-sized humanoid robot JAXON [1], which holds a racket with its right hand. We choose the following parameters: we set the number of the uniform B-spline control points as $m = 14$, the order of the B-splines as $n = 5$, the finish time as $t_f = 2.6$s, and the weight of the objective function as $w = 0.999995$. We also set the limit of the joint velocities and the three key poses, which are the joint angles of the initial pose $\boldsymbol{\theta}_0$, the hitting pose $\boldsymbol{\theta}_h$, and the final pose $\boldsymbol{\theta}_f$ as the values shown in Fig.3 and Table.I . These key poses are computed by using the human motion as reference.



(1) initial pose (2) hitting pose (3) final pose

Fig. 3. Three key poses

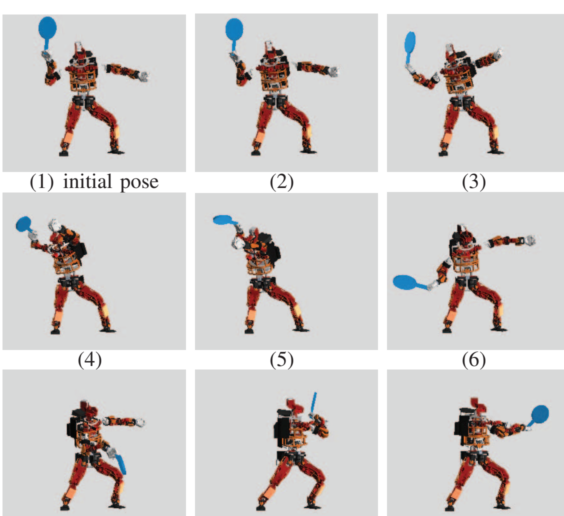
We programmed our algorithm in the Euslisp language [11]. Our method is time-consuming and it takes about a week for the offline swing optimization, although it is probable that our algorithm can compute the solution more rapidly since the Euslisp is an interpreted language.

TABLE I

JAXON's Max Joint Velocities and Key Pose Joint Angles

Limb	Joint	Max Joint velocity [rad/s]	Initial Pose $\boldsymbol{\theta}_0$ [deg]	Hitting Pose $\boldsymbol{\theta}_h$ [deg]	Final Pose $\boldsymbol{\theta}_f$ [deg]
Head	neck-y	4.0		22.40	
	neck-p	4.0		0.00	
Torso	waist-r	4.0	4.67	6.57	2.20
	waist-p	4.0	16.36	21.07	8.55
	waist-y	4.0	-11.29	26.89	46.42
Right arm	collar-y	4.0	-2.74	49.26	57.58
	shoulder-p	4.0	-51.24	2.92	-56.24
	shoulder-r	4.0	-49.00	-35.34	-54.50
	shoulder-y	4.0	2.21	-30.04	32.61
	elbow-p	4.0	-118.21	-105.07	-61.60
	wrist-y	4.0	-29.91	-103.75	-75.42
	wrist-r	4.0	-31.53	-55.44	14.53
	wrist-p	4.0	3.22	-7.25	25.79
Left arm	collar-y	4.0		8.00	
	shoulder-p	4.0		0.00	
	shoulder-r	4.0		80.00	
	shoulder-y	4.0		0.00	
	elbow-p	4.0		-40.00	
	wrist-y	4.0		45.00	
	wrist-r	4.0		0.00	
	wrist-p	4.0		0.00	
Right leg	crotch-y	9.0	5.92	-9.59	-19.14
	crotch-r	9.0	-17.97	-23.05	-26.16
	crotch-p	9.0	-43.65	-40.30	-32.42
	knee-p	9.0	79.70	79.15	68.15
	ankle-p	9.0	-34.99	-36.66	-15.38
	ankle-r	9.0	17.36	25.23	27.03
Left leg	crotch-y	9.0	50.00	33.95	25.85
	crotch-r	9.0	27.60	23.44	20.89
	crotch-p	9.0	-46.13	-54.83	-51.45
	knee-p	9.0	57.43	76.68	73.27
	ankle-p	9.0	-10.00	-20.45	-22.24
	ankle-r	9.0	-27.34	-21.93	-20.05

As a result of the optimization, the hitting time (t_h = 1.596s) and the trajectory parameters are solved. As shown in Fig.4, these parameters result in the dynamic whole-body swing motion including a backswing. Fig.5 shows the right arm joint velocities of the planned motion. Specific joint velocities reach almost at the maximum limit, which means that the robot can utilize its highest physical performance. Fig.6 shows the linear velocity of the racket in a normal



(1) initial pose (2) (3)

(4) (5) (6)

(7) hitting pose (8) (9) final pose

Fig. 4. Overview of the optimal swing motion generated by the offline swing optimization

direction of the racket surface. This indicates that the swing speed reaches up to 14.6m/s.

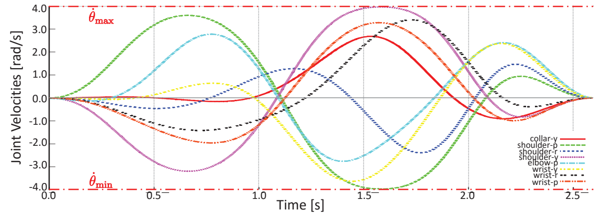


Fig. 5. Right arm joint velocities of the optimal swing motion

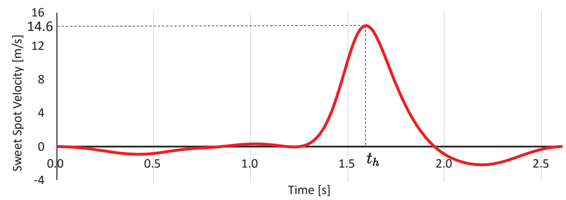


Fig. 6. Linear velocity of the racket (sweet spot) in the $\boldsymbol{n}$ direction

IV. Online Swing Modification

Suppose a humanoid robot hits the flying ball with the optimal swing motion. The predicted ball position at the hitting time $t = t_h$ needs to be accurate when the robot starts its swing motion ($t = 0$). Due to the uncertainty caused by the error of the recognition and motion model of the ball, however, it is much difficult to predict the ball trajectory t_h seconds later at the start time $t = 0$ accurately.

In this paper, therefore, we propose the method of the online swing modification, assuming that the predicted ball position is changed before the hitting time $t = t_h$. The method generates the motion which satisfies the reference hitting pose $\boldsymbol{\theta}_h^{ref}$. The reference hitting pose is updated according to the change of the predicted ball position. To modify swing motion online, we modify the optimized control point vector $\boldsymbol{p}$ with the constraints reduced. By solving a quadratic programming problem, we compute the modification amount which is assumed to be small.

A. Reference Hitting Pose Update

To reduce parameters which need changing, we modify only the racket arm joint trajectories to satisfy the updated reference hitting pose $\boldsymbol{\theta}_h^{ref}$ at the hitting time $t = t_h$. Although the strict optimality is lost, it is possible to modify swing motion online — the step time Δt for the update of the hitting pose is short enough compared with the overall time of swing motion t_f. The reference hitting pose $\boldsymbol{\theta}_h^{ref}$ is calculated by solving inverse kinematics for the racket arm, and the modification amount $\boldsymbol{\Delta\theta}_h$ is computed as follows:

$$\boldsymbol{\Delta\theta}_h = \boldsymbol{\theta}_h^{ref} - \boldsymbol{\theta}_h \quad (16)$$

B. Online Trajectory Modification

When the reference joint angles at a certain time later are changed at the current time, a possible approach is to interpolate trajectories such as Hoff-Arbib interpolation [9], [12]. However, it is likely that joint velocities exceed the limit during the interpolation, since the robot swings fast with the joint velocities reaching almost the limit. This might cause damage to hardware. In this study, we generate joint j's trajectory by adding the control point vector $\boldsymbol{p}_j$ of joint j to the modification amount $\Delta\boldsymbol{p}_j$. We determine the modification amount $\Delta\boldsymbol{p}_j$ with the control point vector $\boldsymbol{p}_j$ satisfying Eq. 12.

Owing to the B-splines' local property, only the control point vector $\boldsymbol{p}_{j,i_h^*}$ regarding the hitting time t_h should be extracted and modified for satisfaction of the reference hitting pose $\boldsymbol{\theta}_{j,h}^{ref}$, regardless of the current time $t_c(< t_h)$. However, this modification might result in the discontinuity of the trajectory $\theta_j(t_c)$ if the current time t_c is close to the hitting time t_h. Thus, we modify not only $\boldsymbol{p}_{j,i_h^*}$ regarding the hitting time t_h but also $\boldsymbol{p}_{j,i_c^*}$ regarding the current time t_c. In other words, we extract the control point vector $\boldsymbol{p}_{j,i^*} = [\boldsymbol{p}_{j,i_c^*}^T \ \boldsymbol{p}_{j,i_h^*}^T]^T \in \mathbb{R}^k$ where k is the dimension of the modification amount of the control point vector $\boldsymbol{\Delta p}_j$. Then we calculate the modification amount of the control point vector $\boldsymbol{\Delta p}_j$, in order to satisfy the continuity at the current time $t = t_c$ and the reference hitting pose at the hitting time $t = t_h$ (Fig.7).

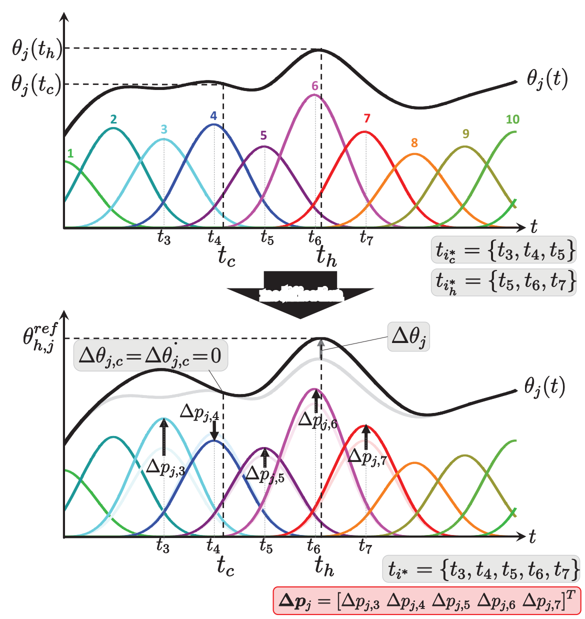


Fig. 7. Trajectory modification using local property of the uniform B-splines

The method for computing the modification amount $\boldsymbol{\Delta p}_j$ is to solve the quadratic programming problem as follows:

$$\min_{\boldsymbol{\Delta p}_j} \boldsymbol{\Delta p}_j^T \boldsymbol{W} \boldsymbol{\Delta p}_j \tag{17}$$

$$\text{subject to} \quad \begin{cases} \Delta\theta_j(t_c) = \boldsymbol{\Delta p}_j \boldsymbol{b}_{i,n}(t_c) = 0 \\ \Delta\dot{\theta}_j(t_c) = \boldsymbol{\Delta p}_j \dot{\boldsymbol{b}}_{i,n}(t_c) = 0 \\ \Delta\theta_j(t_h) = \boldsymbol{\Delta p}_j \boldsymbol{b}_{i,n}(t_h) = \Delta\theta_j \end{cases} \tag{18}$$

$$\dot{\theta}_{\min} - q_i \leq \Delta q_i \leq \dot{\theta}_{\max} - q_i \tag{19}$$

where $\boldsymbol{W}$ is a weight matrix. Eq. 17 is the objective function for minimizing the modification amount as much as possible. The equality constraints written as Eq. 18 are the ones on the continuity of the joint angles and joint velocities at the current time $t = t_c$ and satisfaction of the reference hitting pose at the hitting time $t = t_h$. The inequality constraint represented as Eq. 19 should be satisfied in order not to exceed the limit of the joint velocities. We use the B-splines' convex hull property described in Sec. III-A.

Since we assume the modification amount of the hitting pose $\boldsymbol{\Delta\theta}_h$ to be small, the constraints on the range of the joint angles, collision check and balancing are not taken into account for the purpose of modifying the hitting pose online (by intervals of Δt seconds).

V. Experimental validation

To verify the effectiveness of our method, we carried out the experiment in which the actual robot JAXON executed the optimal swing motion generated by the offline swing optimization. The swing motion is performed under the sensor-feedback system with the ZMP trajectory as a reference (described as "Stabilizer" in Fig.2) [13]. We confirm that JAXON performs the dynamic swing motion successfully without falling (Fig.8). Fig.9 shows that the motion is feasible with regard to the joint torques, although we do not include the inequality constraints on the joint torque limits.

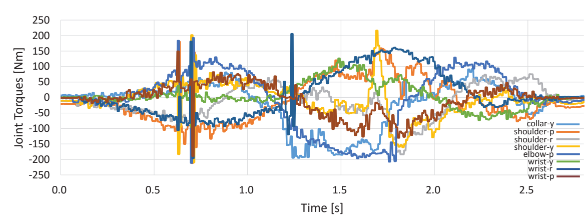


Fig. 9. Right arm joint toruques of the optimal swing motion

We also conducted the experiments of modifying swing motion in accordance with the change of the predicted ball position in the simulation. The predicted ball position is equivalent to the target ball position in this context. The change of the target position was programmed, and the robot could know the exact position in this experiment. Assuming that the robot hit the ball at some point in the cross section hyperplane (the XZ-plane), we tested the target ball position changing from the original position at a constant speed in one direction ($\pm x$ direction and $\pm z$ direction) as illustrated in Fig.10. We set the weight matrix as $\boldsymbol{W} = diag(1, 2, \cdots, k)$, and the step time as $\Delta t = 0.065$s. We continued to update the reference hitting pose until shortly before the hitting time t_h, which was $t = 1.43$s.

Fig.11 shows the overview of the experiment of the online swing modification under the condition of $\boldsymbol{\Delta r} = [0\ 0\ -0.2]^T$m/s. As a result of the experiments, the robot was able to perform swing motion without falling down or self-collision as long as $\boldsymbol{\Delta r}$ was $[0.05\ \ 0\ \ 0]^T$m/s or $[-0.02\ \ 0\ \ 0]^T$m/s in x direction, and $[0\ \ 0\ \ 0.06]^T$m/s or $[0\ 0\ -0.2]^T$m/s in z direction. However, the robot was apt to fall down or detect self-collision if $\boldsymbol{\Delta r}$ exceeded these

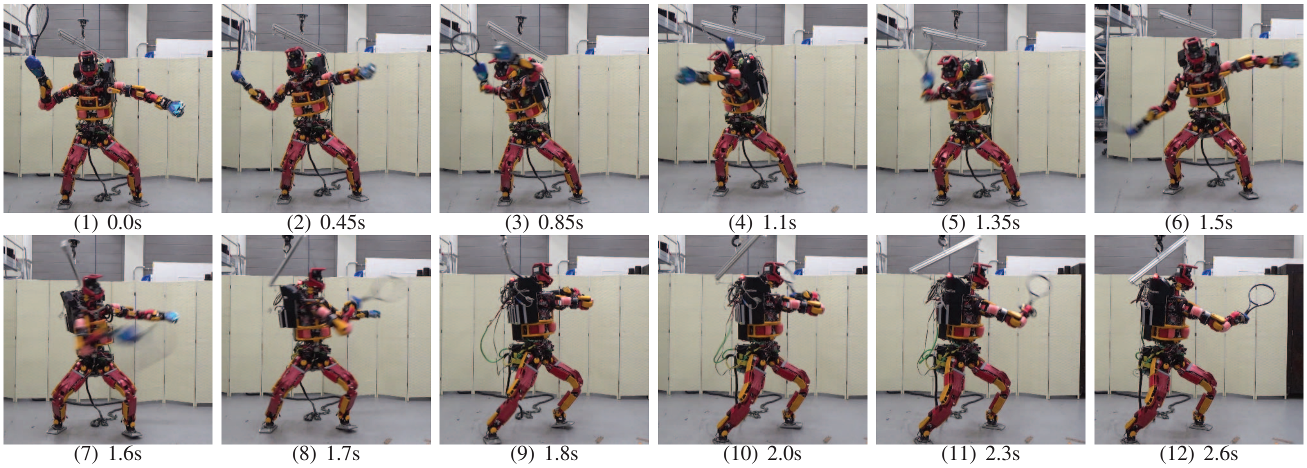


Fig. 8. Snapshot of the optimal swing motion performed by JAXON

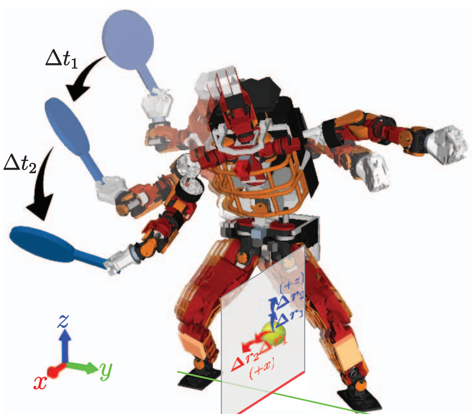


Fig. 10. Experimental conditions for the change of the target ball position

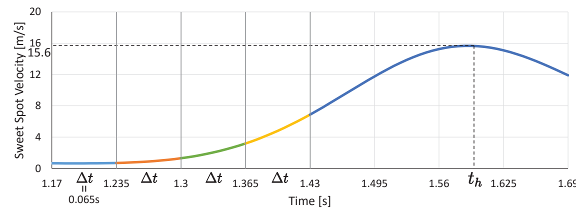


Fig. 13. Planned linear velocity of a racket in a normal direction of a racket surface under $\boldsymbol{\Delta r} = [0\ 0\ \ -0.2]^T$m/s at 1.17s $< t <$ 1.69s; the swing speed is changed acoording to the modificed swing motion which is succesively modified by the step time Δt.

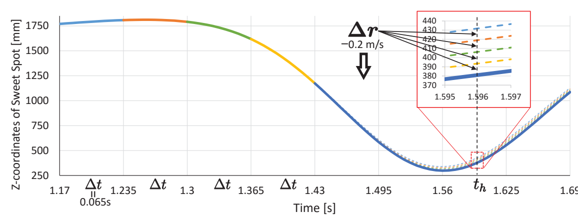


Fig. 14. Temporal change of the racket z coordinates under $\boldsymbol{\Delta r} = [0\ 0\ -0.2]^T$m/s at 1.17s $< t <$ 1.69s; the z-coordinates of the sweet spot at the hitting time t_h is succesively modified, following the target position which moves at the rate of $\boldsymbol{\Delta r}$

values. These results mean that the robot is able to modify more in z direction than in x direction, and the most in $-z$ direction.

Fig.13 shows the planned linear velocity of the racket in a normal direction of the racket surface under $\boldsymbol{\Delta r} = [0\ 0\ -0.2]^T$m/s. This indicates that the peak swing speed is 15.6m/s. This speed is above the speed of the optimal swing motion since the moment arm is longer, but the proposed method hardly loses the optimality even when the hitting pose is modified. Fig.14 shows the temporal change of the racket z coordinates under $\boldsymbol{\Delta r} = [0\ 0\ \ -0.2]^T$m/s. This graph shows that the racket position at the hitting time t_h sequentially planned by the step time Δt is following the target position which is moving at the rate of $\boldsymbol{\Delta r} = [0\ 0\ \ -0.2]^T$m/s. This means that the proposed method can modify the racket position to the extent of -0.286m (≈ -0.2m/s$\times$1.43s) in the z axis in this swing motion.

Provided that the original predicted position differs, the probable approach is to prepare the patterns of the swing motion corresponding to the various target ball positions by the offline swing optimization. The pattern should be chosen at the start time $t = 0$ according to the original target. Suppose the other patterns of the swing motion also have a room for modification to the same extent as experimental values in the worst case ($\pm$0.02m/s in x direction and $\pm$0.06m/s in z direction). We should prepare the optimal swing motion patterns at least by about 0.05m ($\approx$ 0.02m/s$\times$1.43s$\times$2) $\times$ 0.17m ($\approx$ 0.06m/s$\times$1.43s$\times$2) grid. If each swing pattern takes charge of the target position in each grid, the required accuracy of the predicted ball position is almost equivalent to the range of this grid. This indicates that the accuracy within about 0.05m$\times$0.17m grid is needed t_h seconds before the robot hits the ball.

However, the swing motion performed by both the actual robot and the robot in the simulation are unstable with its soles slipping and its toes or heels lifting off. It could be a big problem on the assumption of hitting the ball, since the position of the racket in Cartesian space is changed if the robot slips before the hitting time t_h. Accordingly, the

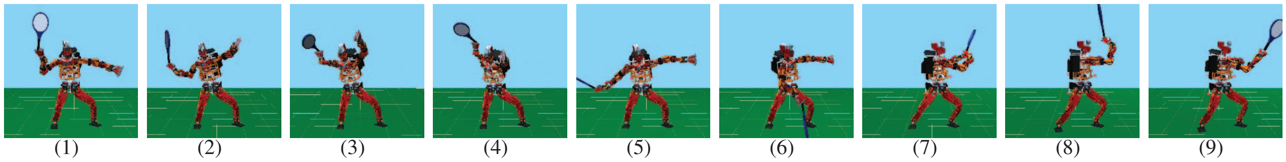


Fig. 11. Snapshot of the experiment of the online swing modification under $\boldsymbol{\Delta r} = [0\ 0\ \ -0.2]$m/s

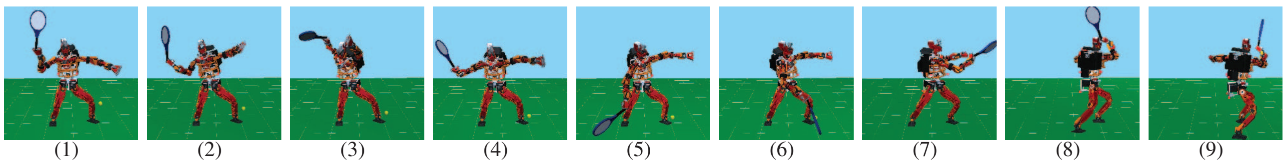


Fig. 12. Snapshot of the experiment of the online swing modification with hitting the ball under $\boldsymbol{\Delta r} = [0\ 0\ \ -0.2]$m/s

yaw momentum compensation (for example, [14]) or the conditions for the friction of the robot soles (for example, [15]) should be taken into account in both the offline swing optimization and online swing modification in the future. Otherwise, the robot should take a step forward with the right foot.

In addition, our method does not take into account the reaction force of the ball. We conducted the experiment of hitting the ball in swing modification under $\boldsymbol{\Delta r} = [0\ 0\ \ -0.2]^T$m/s, setting the tennis ball in the simulation (Fig.12). Compared with the swing motion without hitting the ball (Fig.11), the swing motion with hitting the ball is less stable due to the reaction force of the ball. Therefore, the method for generating swing motion online considering the balance of the robot should be proposed to extend the modification amount (also left for future work).

VI. Conclusion

In this paper, in order to achieve higher physical performance of a humanoid robot, we propose the methods of the offline swing optimization and the online swing modification for generating dynamic tennis forehand swing motion. The method of the offline swing optimization maximizes the swing speed and satisfies the constraints by means of optimizing the uniform B-spline control points which construct joint trajectories. The method of the online swing modification modifies a part of the optimal swing motion online considering joint velocity limits according to the updated hitting pose. These methods are verified through the experiments. We confirm that the offline swing optimization generates the feasible motion for a humanoid robot without falling down or self-collision. We also confirm that the online swing modification can modify the swing motion successively in accordance with the change of the predicted ball position in the simulation. Then we evaluate the method of the online swing modification in terms of the required accuracy of the predicted ball position.

References

[1] K. Kojima, T. Karasawa, T. Kozuki, E. Kuroiwa, S. Yukizaki, S. Iwaishi, T. Ishikawa, R. Koyama, S. Noda, F. Sugai, S. Nozawa, Y. Kakiuchi, K. Okada, and M. Inaba, "Development of life-sized high-power humanoid robot JAXON for real-world use," in *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*, Nov. 2015, pp. 838–843.

[2] Y. Sun, R. Xiong, Q. Zhu, J. Wu, and J. Chu, "Balance motion generation for a humanoid robot playing table tennis," in *Proceedings of the 2011 IEEE-RAS International Conference on Humanoid Robots*, Oct. 2011, pp. 19–25.

[3] Z. Yu, Y. Liu, Q. Huang, X. Chen, W. Zhang, J. Li, G. Ma, L. Meng, T. Li, and W. Zhang, "Design of a humanoid ping-pong player robot with redundant joints," in *Proceedings of the 2013 IEEE International Conference on Robotics and Biomimetics*, Dec. 2013, pp. 911–916.

[4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, vol. 2, May 2002, pp. 1398–1403.

[5] S. Hyon, J. Moren, and G. Cheng, "Humanoid batting with bipedal balancing," in *Proceedings of the 2008 IEEE-RAS International Conference on Humanoid Robots*, Dec. 2008, pp. 493–499.

[6] E. Demircan, T. F. Besier, and O. Khatib, "Muscle force transmission to operational space accelerations during elite golf swings," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1464–1469.

[7] C. de Boor, "On calculating with b-splines," *Journal of Approximation Theory*, vol. 6, pp. 50–62, 1972.

[8] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.

[9] B. Hoff and M. A. Arbib, "Models of trajectory formation and temporal interaction of reach and grasp," *Journal of Motor Behavior*, vol. 25, no. 3, pp. 175–192, 1993.

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

[11] T. Matsui and M. Inaba, "Euslisp: An object-based implementation of lisp," *Journal of Information Processing*, vol. 13, no. 3, pp. 327–338, 1990.

[12] S. Nozawa, E. Kuroiwa, K. Kojima, R. Ueda, M. Murooka, S. Noda, I. Kumagai, Y. Ohara, Y. Kakiuchi, K. Okada, and M. Inaba, "Multi-layered real-time controllers for humanoid's manipulation and locomotion tasks with emergency stop," in *Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots*, Nov. 2015, pp. 381–388.

[13] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 4489–4496.

[14] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Resolved momentum control: humanoid motion planning based on the linear and angular momentum," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Oct. 2003, pp. 1644–1650.

[15] Y. Tamiya, M. Inaba, and H. Inoue, "Realtime balance compensation for dynamic motion of full-body humanoid standing on one leg," *Journal of the Robotics Society of Japan*, vol. 17, no. 2, pp. 268–274, 1999.