2012 12th IEEE-RAS International Conference on Humanoid Robots
Nov.29-Dec.1, 2012. Business Innovation Center Osaka, Japan

Kinematic Optimization and Online Adaptation of Swing Foot Trajectory for Biped Locomotion

Hiroshi Kaminaga
Graduate School of Information Science and Technology
The University of Tokyo
Tokyo, Japan 113–8656
Email: kaminaga@ynl.t.u-tokyo.ac.jp

Johannes Englsberger and Christian Ott
Institute of Robotics and Mechatronics
German Aerospace Center (DLR)
Weßling, Germany 82234

***Abstract*—Biped locomotion involves many degree of freedom (DOF). Some of the DOFs can be decided upon objectives of the locomotion and balance maintenance. However, there are often redundant DOFs, which are decided with heuristic criteria. On the other hand, walking speed is often limited by the maximum permissible joint speed, which is affected by the given heuristic trajectory. In this paper, we propose a kinematic optimization method to reduce maximum joint speed for enhancing mobility of the locomotion. The method was evaluated with experiment.**

I. Introduction

Biped robots have a large number of actuated joints to enable versatile motion of their body in order to maintain their posture during locomotion. Typically they have more than 6 joints per leg to enable manipulation of the CoM (Center of Mass) that determines approximate position of CoP (Center of Pressure) in the supporting region.

Biped locomotion is a type of legged locomotion that uses a pair of limbs. It is a minimal configuration of legged mechanism that can walk. Biped locomotion has the least spatial requirement for legged locomotion but it requires sophisticated control to maintain the balance while walking due to its non holonomic nature. Ultimately, biped mechanisms have advantages when the robot is required to navigate through an environment that originally was designed only for human.

Locomotion control of a biped robot is essentially a control of GRF (Ground Reaction Force) through the control of the CoM position and angular momentum around the CoM [1]. To have control over both CoM and swing leg, biped robots are usually equipped with 6DOF (Degree of Freedom) per leg resulting at least 12 DOF for lower body. With two 6 DOF legs, both CoM and swing foot can be placed arbitrary if the target position is achievable within the range of motion of joints. Considering that the industrial manipulators usually have 6 or 7 DOF, biped robots require control of large number of joints, even when only lower body was considered. Trajectory planning of such large number of joints is not a trivial problem.

In modern biped robots, the dynamics is often simplified to a point mass at the CoM of the robot [2], [3]. This simplification reduces the motion planning down to 3 DOF that is far less than the case of the full model that has 6+N DOF where N is the number of joints. That leaves 3DOF redundancy on support leg trajectory planning and undetermined 6DOF on the swing leg. Specifying geometric trajectory of the swing leg still leaves infinite degree of freedom when the temporal trajectory is considered. Giving heuristic constraints, or predefined spatiotemporal trajectory enables us to resolve the redundancy, but it will give unnecessary constraint that can limit the robot's performance.

One of the way to make use of the redundancy is to use it to shape the robot dynamics. This enables us to have more control over the postural balancing while executing the same foot positioning. Optimization is often used to obtain such dynamic trajectory. However, the optimization takes time and the optimum trajectory is both time and space dependent.

In this paper, our objective is to propose a method to generate swing foot trajectory by kinematic optimization considering joint speed. Since the maximum actuator torque and speed have some fixed relationship, having smaller joint speed for same task leads to more permissible torque, hence higher mobility performance. Since the optimization becomes nonlinear, this process is time consuming and thus it is unrealistic to prepare all possible gait if the optimization is performed off-line. We chose to prepare limited set of gait as optimized trajectory, and deform this trajectory to adapt to desired boundary conditions. Null space mapping of a trajectory function was adopted to preserve fundamental characteristics of the curve and still satisfy acceleration continuity and boundary condition such as target foot position.

II. Kinematic Structure of DLR Biped

The DLR Biped is a lower body biped robot developed by German Aerospace Center shown in Fig. 1 [4], [5].

The robot is equipped with 12 DOF (6 on each limb) with joint configuration shown in Fig. 2. The method presented in this paper was applied to DLR biped. However, the method is general and is not restricted to the structure of the robot.

III. Description of Swing Foot Trajectory

Let us assume a function of time $\boldsymbol{f}(t, \phi) \in \boldsymbol{R}^n$ that gives a trajectory of some variable of dimension n. Let us assume

978-1-4673-1369-8/12/$31.00 ©2012 IEEE

Fig. 1. The DLR Biped

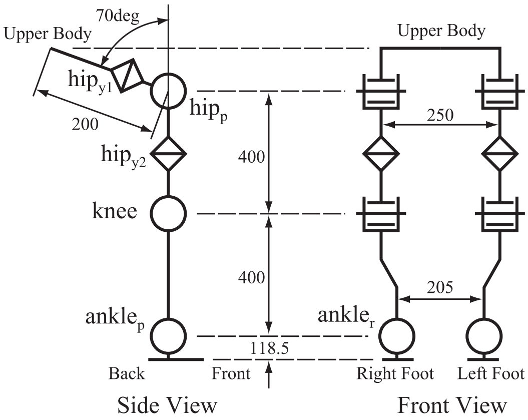


Fig. 2. Joint configuration of DLR Biped. Units are in (mm)

that the function $\boldsymbol{f}$ is parameterized with a parameter vector $\boldsymbol{\phi} \in \boldsymbol{R}^N$ where N is the dimension of the parameter vector.

When considering the trajectory optimization of lower body movement of locomotion, spatiotemporal condition needs to be considered. Hence, boundary conditions are specified with position, its time derivatives, and time of the event. In this case, we can assume $\boldsymbol{\phi}$ to be consisting of $(t_i, \boldsymbol{f}(t_i))$, $(t_i, \boldsymbol{f}^{(1)}(t_i))$, $\cdots$, $(t_i, \boldsymbol{f}^{(s)}(t_i))$ where t_i is a time of the control point and $\boldsymbol{f}^{(s)}$ shows s-th time derivative of a function $\boldsymbol{f}$.

A m-th order scalar polynomial function can be expressed in the following form.

$$f(t) = a_m t^m + a_{m-1} t^{m-1} + \cdots + a_1 t + a_0 \tag{1}$$

Let us express this function with a coefficient vector $\boldsymbol{a} = \begin{bmatrix} a_m & a_{m-1} & \cdots & a_1 & a_0 \end{bmatrix}^T$ using the form

$$f(t) = \boldsymbol{t}(t)^T \boldsymbol{a} \tag{2}$$

where

$$\boldsymbol{t}(t) = \begin{bmatrix} t^m & t^{m-1} & \cdots & t & 1 \end{bmatrix} \tag{3}$$

.

Using a matrix D, a s-th order time derivative of f is given as follows.

$$f^{(s)}(t) = (D^s \boldsymbol{t}(t))^T \boldsymbol{a} \tag{4}$$

where

$$D = \begin{bmatrix} 0 & m & 0 & \cdots & 0 \\ 0 & 0 & m-1 & \cdots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \in \boldsymbol{R}^{(m+1)\times(m+1)} \tag{5}$$

Coefficient vector $\boldsymbol{a}$ satisfying boundary conditions given with time t_i and s_i-th time derivative at t_i, can be calculated as follows.

$$\boldsymbol{a} = T^{-1} Y \tag{6}$$

where

$$T = \begin{bmatrix} D^{s_1}\boldsymbol{t}(t_1) & D^{s_2}\boldsymbol{t}(t_2) & \cdots & D^{s_{m+1}}\boldsymbol{t}(t_{m+1}) \end{bmatrix}^T \tag{7}$$

and

$$Y = \begin{bmatrix} f^{(s_1)}(t_1) & f^{(s_2)}(t_2) & \cdots & f^{(s_{m+1})}(t_{m+1}) \end{bmatrix}^T \tag{8}$$

When there are more boundary conditions given or less boundary condition given, pseudoinverse can be used instead of matrix inverse to best approximate, or resolve the parameter redundancy in the sense of least squares.

In this type of formulation, parameter vector $\boldsymbol{\phi}$ can be expressed as a vector of boundary conditions as follows.

$$\boldsymbol{\phi} = \left[t_1, \cdots, t_{m+1}, f^{(s_1)}(t_1), \cdots, f^{(s_{m+1})}(t_{m+1})\right]^T \tag{9}$$

In this paper, we express the motion of swing foot as translational position x_e, y_e, z_e and X-Y-Z Euler angles $\alpha_e, \beta_e, \gamma_e$. For an example we assume x_e to be expressed as $x_e = x_e(t, \boldsymbol{\phi}_{xe})$. We put a superscript s as ${}^s x$ when the parameter is expressed with respect to the support foot frame, which is located at the center of the sole of the support foot, x pointing toward the front, z to the vertical up. We describe the motion with these 6 parameters with independent 6 polynomial curves. We chose center of toe tip as the end-effector position. The reason we did not perform the optimization for all the joint angles was reduction of the optimization parameters. If we try to optimize all polynomials for all the joints, the optimization will have too many local minima.

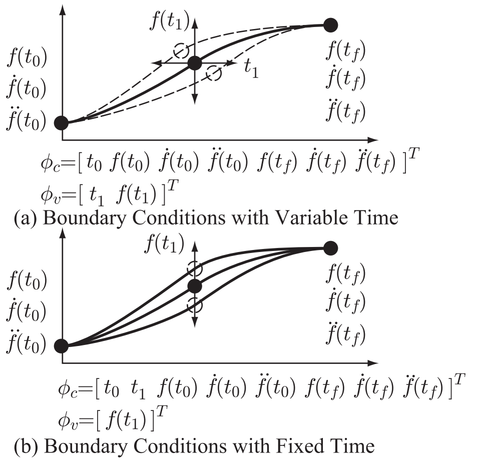
Similarly, we express hip position of the robot with translational position x_h, y_h, z_h and X-Y-Z Euler angles $\alpha_h, \beta_h, \gamma_h$. We assume these parameters being expressed also with polynomial functions.

IV. Trajectory Optimization

The optimization of a trajectory is to find a optimal $\boldsymbol{\phi}$ as follows:

$$\boldsymbol{\phi}^* = \underset{\boldsymbol{\phi}}{\operatorname{argmin}}\, J(\boldsymbol{f}(t, \boldsymbol{\phi})) \text{ for } \in [0, t_f] \tag{10}$$

The cost function may chosen to take the integral of the cost over the defined period as in quadratic programming, or take



ϕ_c=[t_0 $f(t_0)$ $\dot{f}(t_0)$ $\ddot{f}(t_0)$ $f(t_f)$ $\dot{f}(t_f)$ $\ddot{f}(t_f)$ $]^T$
ϕ_v=[t_1 $f(t_1)$ $]^T$
(a) Boundary Conditions with Variable Time

ϕ_c=[t_0 t_1 $f(t_0)$ $\dot{f}(t_0)$ $\ddot{f}(t_0)$ $f(t_f)$ $\dot{f}(t_f)$ $\ddot{f}(t_f)$ $]^T$
ϕ_v=[$f(t_1)$ $]^T$
(b) Boundary Conditions with Fixed Time

Fig. 3. Concept of Boundary Condition Parameter Vectors

maximum of the period. Usually optimization problem has some equality and inequality constraints generally written in the following form with constraint function C_i and constraint set S_i.

$$C_i(\boldsymbol{\phi}) \in S_i \subset \boldsymbol{R}, i = 1, 2, \ldots, n_c \tag{11}$$

When considering trajectory optimization of a gait, initial and final position, velocity, and acceleration are predefined. There are also some additional parameters as stepping period and step height (maximum height of the swing foot during the swing phase) that are pre-defined. Hence, some elements of $\boldsymbol{\phi}$ are predefined constant and some are parameters that can be altered with optimization. Let us put constant portion of $\boldsymbol{\phi}$ as $\boldsymbol{\phi}_c$ and variable portion $\boldsymbol{\phi}_v$. Thus $\boldsymbol{\phi} = (\boldsymbol{\phi}_c, \boldsymbol{\phi}_v)$. It should be noted that elements in $\boldsymbol{\phi}_c$ and $\boldsymbol{\phi}_v$ can be chosen arbitrary. In extreme cases, there can be cases that all time parameters t_i to be constants or all t_i to be variables. The concept of $\boldsymbol{\phi}_c$ and $\boldsymbol{\phi}_v$ is shown in Fig. 3.

Using notation above, (10) can be written as follows.

$$\boldsymbol{\phi} = \left(\boldsymbol{\phi}_c, \underset{\boldsymbol{\phi}_v}{\operatorname{argmin}} J(\boldsymbol{f}(t, \boldsymbol{\phi})) \right) \tag{12}$$

Our strategy is to minimize maximum joint speed to enhance mobility of the robot. Thus, we chose following cost function as the optimization criteria.

$$J = \|\dot{\boldsymbol{q}}^*(t)\|_\infty \tag{13}$$

Here, $\dot{\boldsymbol{q}}^*$ is the vector of the joint speed normalized by its maximum permissible speed. $\dot{\boldsymbol{q}}$ can be calculated from desired trajectory of the swing foot by using differential inverse kinematics. *i.e.* putting $\nabla_q \boldsymbol{r} = \partial \boldsymbol{r} / \partial \boldsymbol{q}$ as the Jacobian matrix of the end-effector with regarding to the joint angles, $\dot{\boldsymbol{q}} = (\nabla_q \boldsymbol{r})^\# \dot{\boldsymbol{r}}$. Here, $(\cdot)^\#$ shows a pseudoinverse of a matrix. Since the relation between optimization parameters and $\dot{\boldsymbol{q}}$ is nonlinear, the problem becomes nonlinear optimization.

There are also nonlinear constraints that avoids the foot to get in contact with the floor or the stance leg, and limits the joint speed in the range of permissible joint speed [5].

V. Online Trajectory Deformation for Adaptation

Although the path acquired above is optimal in the sense of the maximum joint speed, planned gait and what must be executed differs due to the feedback signals for stabilization; robot reacts to perturbations and corrects CoM position and velocity. Also, if the robot changes the gait, optimization must be performed again for the gait with appropriate stride. However, it is unrealistic to calculate optimal trajectories for all possible gait patterns. Moreover, such method cannot handle the transition between different gaits. The best solution to this would be to optimize the trajectory online to take in account of this perturbation, but in reality, nonlinear optimization is too slow to be executed online.

To adapt to the real situation, our strategy is to deform the existing trajectory to path satisfying real situation. The generated path would not be truly optimal, but when the deformation is small, we can assume the path to be near-optimal.

The criteria for the deformation is as follows:

1) The path is acceleration continuous
2) The path satisfies arbitrary boundary conditions (current state and the terminal condition)
3) The path preserves similarity to the curve form of the original path
4) The path regeneration can be done every time-step

In this paper, we chose the coefficient vector $\boldsymbol{a}$ of a polynomial to represent function space of a trajectory. We define the distance between polynomial by the Euclidian distance between coefficient vectors. We assume that if the difference in coefficients are small, form of the curve is preserved since coefficients encapsule higher order derivative of the curve.

This assumption makes the problem of trajectory deformation as the problem of finding a coefficient vector $\boldsymbol{a}'$ that is close to $\boldsymbol{a}$ while satisfying boundary conditions.

Boundary conditions we want to satisfy are as follows:

1) Current position $f(t)$
2) Current velocity $\dot{f}(t)$
3) Current acceleration $\ddot{f}(t)$
4) Terminal position $f(t_f)$
5) Terminal velocity $\dot{f}(t_f)$
6) Terminal acceleration $\ddot{f}(t_f)$

The polynomial that satisfies the conditions can be calculated with following calculation.

$$\boldsymbol{a}' = \hat{T}^{-1} \hat{Y} \tag{14}$$

where

$$\begin{aligned} \hat{T} &= [\ \boldsymbol{t}(t) \quad D\boldsymbol{t}(t) \quad D^2\boldsymbol{t}(t) \\ &\qquad \boldsymbol{t}(t_f) \quad D\boldsymbol{t}(t_f) \quad D^2\boldsymbol{t}(t_f)\ ]^T \end{aligned} \tag{15}$$

$$\begin{aligned} \hat{Y} &= [\ f(t) \quad \dot{f}(t) \quad \ddot{f}(t) \\ &\qquad f(t_f) \quad \dot{f}(t_f) \quad \ddot{f}(t_f)\ ]^T \end{aligned} \tag{16}$$

If the polynomial order is larger than 5, there will be redundant space in the coefficient vector and (14) becomes as (17).

$$\hat{\boldsymbol{a}} = \hat{T}^{\#}\hat{Y} + \left(I - T^{\#}T\right)\boldsymbol{\xi} \tag{17}$$

where $\boldsymbol{\xi}$ is arbitrary vector with appropriate size. The second term of (17) shows the null-space of the boundary conditions. Based on the idea of Nakamura et al. [6], we utilize this null space to satisfy the boundary condition and make best choice of coefficients close to original ones to preserve form of the original polynomial curve.

First, we pad 0 to increase the order of the polynomial as $\hat{\boldsymbol{a}} = \left[\begin{array}{cc} \boldsymbol{0} & \boldsymbol{a}^T \end{array}\right]^T$. This is necessary that because if the order of the polynomial is less than equal to 5, there is no more room left for preserving the characteristics of the original trajectory. Using this coefficient vector, we calculate new coefficient vector $\boldsymbol{a}'$ that satisfies boundary conditions as follows.

$$\boldsymbol{a}' = \hat{T}^{\#}\hat{Y} + \left(I - T^{\#}T\right)\hat{\boldsymbol{a}} \tag{18}$$

In the trajectory generation, this coefficient vector is used instead of $\boldsymbol{a}$.

When the modification of the stepping period is necessary, a matrix M gives the stretch or the shortening of the time.

$$M = \begin{bmatrix} \left(t_f/t_f'\right)^{m+1} & 0 & \cdots & 0 \\ 0 & \left(t_f/t_f'\right)^{m} & \cdots & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{19}$$

Here, t_f' is the new time for the end of the swing phase. Using M, new coefficient vector is given as follows.

$$\boldsymbol{a}' = \hat{T}^{\#}\hat{Y} + \left(I - T^{\#}T\right)M\hat{\boldsymbol{a}} \tag{20}$$

Since this method is done every time-step, trajectory modification is possible within the swing phase that ultimately enables us to change landing position and landing time of the swing foot after the swing foot has taken off.

Fig. 4 shows the trajectory comparison of different adapting method. From the figure, we can see that the trajectory is acceleration continuous and satisfies boundary conditions. In the case of simple stretching, the acceleration continuity of the trajectory is not guaranteed. In the case of Fig. 4, there is a jump in the trajectory at time 0.3s.

VI. Experiment

A. Basic Strategy

In this paper, we focus on the effect of the optimization on the walking performance. Currently, the DLR biped is position controlled and uses capture point [7] as the reference, with force controller for the stabilization as presented in Englsberger et al. [8]. Our strategy is to design a reference optimal foot trajectory and adapt it to the desired conditions.

In [8], the swing foot trajectory was given with a heuristic polynomial function. We want to calculate an optimal trajectory from this original pattern generator. Optimization was done only for cyclic walking pattern of the straight forward walking.

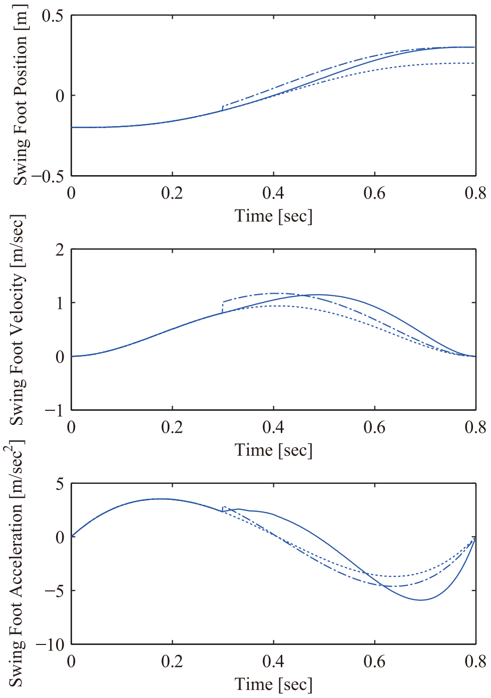


Fig. 4. Trajectory Adaptation Comparison. Target position was changed from 0.2 to 0.3 at 0.3 sec. Dotted line shows original trajectory. Dotted dashed line shows the simple stretch of the trajectory to satisfy terminal condition. Solid line shows proposed method.

The flow of the experiment is as follows.

1) Generate joint angle trajectory of the original pattern generator.
2) Calculate CoM and position of the toe.
3) Extract boundary conditions, such as initial position, final position, step height, from the toe trajectory acquired above.
4) Calculate optimal trajectory satisfying boundary conditions and constraints.
5) Deform optimal trajectory in realtime to satisfy actual boundary conditions including effect of feedback signals

Steps 1 to 4 are performed off-line. In our environment (Intel Core-i7 L620 2.0 GHz, 4GB RAM, MATLAB 2010a), this process takes more than 10 minutes.

In following sections, evaluation of the two types of trajectory optimization is carried out. For the optimization, optimization toolbox of MATLAB (product of MathWorks) [9] was used. Sequential quadratic programming solver was used for optimization of nonlinear cost function and nonlinear constraints.

Effect of changing swing foot trajectory would result in different torso movement. However, in this paper, this variation in torso movement was neglected in the optimization process.

B. Fixed Body Yaw Angle

First, we tested the optimization with the body angle fixed. Hence, only the swing leg is used for optimization. Following conditions were used for the optimization.

$$\begin{aligned}
\boldsymbol{\phi}_{exc} &= [t_0, t_f, x_e(t_0), \dot{x}_e(t_0), x_e(t_f), \dot{x}_e(t_f)]^T \\
\boldsymbol{\phi}_{exv} &= [t_{1x_e}, x_e(t_{1x_e})]^T \qquad (21) \\
\boldsymbol{\phi}_{eyc} &= [t_0, t_{1y_e}, t_f, y_e(t_0), \dot{y}_e(t_0), \ddot{y}_e(t_0), \\
&\quad y_e(t_f), \dot{y}_e(t_f), \ddot{y}_e(t_f)]^T \\
\boldsymbol{\phi}_{eyv} &= [y_e(t_{1y_e})] \qquad (22) \\
\boldsymbol{\phi}_{ezc} &= [t_0, t_f, z_e(t_0), \dot{z}_e(t_0), z_e(t_f), \dot{z}_e(t_f)]^T \\
\boldsymbol{\phi}_{ezv} &= [t_{1z_e}, z_e(t_{1z_e})]^T \qquad (23) \\
\boldsymbol{\phi}_{e\alpha c} &= [t_0, t_{1\alpha_e}, t_f, \alpha_e(t_0), \dot{\alpha}_e(t_0), \alpha_e(t_{1\alpha_e}), \dot{\alpha}_e(t_{1\alpha}), \\
&\quad \alpha_e(t_f), \dot{\alpha}_e(t_f)]^T \\
\boldsymbol{\phi}_{e\alpha v} &= [\ ]^T \qquad (24) \\
\boldsymbol{\phi}_{e\beta c} &= \left[t_0, t_f, \beta_e(t_0), \dot{\beta}_e(t_0), \beta_e(t_f), \dot{\beta}_e(t_f)\right]^T \\
\boldsymbol{\phi}_{e\beta v} &= [t_{1\beta_e}, \beta_e(t_{1\beta_e})]^T \qquad (25) \\
\boldsymbol{\phi}_{e\gamma c} &= [t_0, t_{1\gamma_e}, t_f, \gamma_e(t_0), \dot{\gamma}_e(t_0), \gamma_e(t_{1\gamma_e}), \dot{\gamma}_e(t_{1\gamma}), \\
&\quad \gamma_e(t_f), \dot{\gamma}_e(t_f)]^T \\
\boldsymbol{\phi}_{e\gamma v} &= [\ ]^T \qquad (26)
\end{aligned}$$

We chose to optimize time and position of the middle point of the trajectory for x_e, z_e, β_e. By optimizing β_e, we may benefit from the toe-off motion for reducing joint speed. We chose to only optimize position of the middle point for y_e. We chose to set values to 0 all the time for α_e and γ_e to avoid possible collision of the swing foot to the support leg. t_{1y_e} was set to $t_f/2$ to reduce the optimization cost. We limited the number of optimization parameters since larger optimization parameters often lead to failure of the convergence due to local minima. Stepping period is 0.8 sec and the stride was 0.2 m.

The obtained swing foot trajectory is shown in Fig. 5. From this figure, we can see that the y direction movement is used to distribute the joint movement. Table I shows the maximum permissible joint speed of the robot. Fig. 6 shows the joint speed trajectory of a swing leg. From this figure we can observe the reduction of maximum joint speed. The result shows 6.8% reduction of the cost function (see (13)). The trajectory shows that the robot is using joints pointing in Y direction to distribute joint speed. It is interesting to find that the toe-off motion did not emerge form the optimization with proposed criteria. It implies that the toe-off motion of human comes from different criteria, such as maximizing double support phase period.

On the hardware, maximum stride possible used to be 0.25m with stepping time 0.7sec. Using the optimization, the stride could be extended to 0.35m with stepping time 0.7sec stably. The attached video shows that the robot looses balance with the original trajectory due to the lack of joint speed; robot could not reach the desired posture in the specified time. With the optimized trajectory, robot walks stably. Fig. 7 shows the desired and real trajectory of the walking with 0.35m stride 0.7sec period. From the figure, we can see that the real value follows the desired value, and the walking was stable.

TABLE I
MAXIMUM PERMISSIBLE JOINT SPEED OF DLR BIPED IN RPM

Joint	Value (RPM)
$\mathrm{hip_{y2}}$	30
$\mathrm{hip_{y1}}$, $\mathrm{hip_p}$, and $\mathrm{ankle_r}$	20
knee and $\mathrm{ankle_p}$	18

However, due to the joint angle limit of the ankle pitch joint, the robot was too close to the kinematic limit for further extension of the stride.

C. Utilizing Body Yaw Angle

Next, we chose to use body yaw angle as additional degree of freedom for optimization. This time we fixed the body translational position, roll angle, and pitch angle and used joint angles of both legs for optimization. Additional conditions were introduced on top of the conditions for the fixed body angle case.

$$\begin{aligned}
\boldsymbol{\phi}_{h\beta c} &= [t_0, t_{1\beta_h}, t_f, \beta_h(t_{1\beta_h})]^T \\
\boldsymbol{\phi}_{h\beta v} &= [\beta_h(t_0), \beta_h(t_f)]^T \qquad (27)
\end{aligned}$$

$t_{1\beta_h}$ was set to $t_f/2$ and $\beta_h(t_{1\beta_h})$ was set to 0, thus body facing front in the middle of the swing phase.

To generate continuous yaw angle trajectory, a linear equality constraint was introduced. This constraint ensured the starting angle and final angle of the body yaw to be same magnitude with opposite sign.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \boldsymbol{\phi}_{h\beta} = 0 \qquad (28)$$

The joint speed trajectory for the case with body yaw angle is shown in Fig. 8. The optimization reduced the cost function (13) by 18%. On the hardware, as in the case of fixed body angle, stride could be extended to 0.35m with stepping time of 0.7sec. Fig. 9 shows the desired and real trajectory of the walking with 0.35m stride 0.7sec period. From the figure, we can see that the real value follows the desired value, and the walking was stable. The scene of the experiment is attached as a video. Further extension was not possible due to the slippage between the floor and the robot sole. Is is expected to occur due to the larger angular momentum of the body from body turn. When the robot is equipped with upper body, this yaw angular momentum could be compensated that is expected to further enhance the mobility of the robot.

VII. CONCLUSION

The objective of this paper was to propose an optimization method of swing foot trajectory for maximizing the mobility of a biped robot. Trajectory defined as polynomials

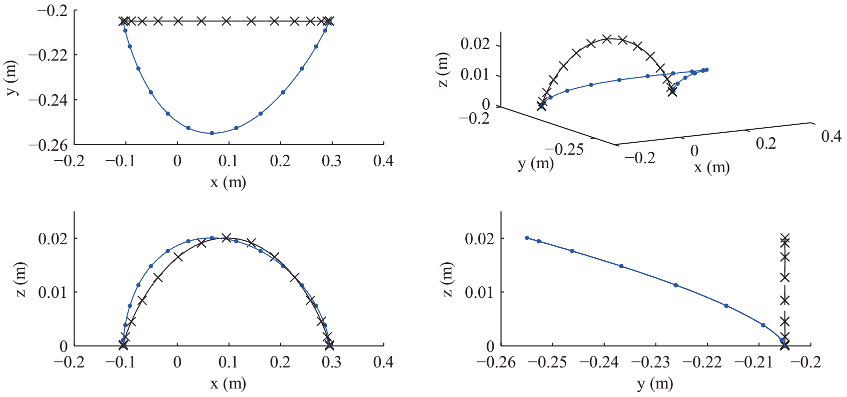


Fig. 5. Swing Foot Trajectory for Fixed Body Angle Case. Black line shows original trajectory and the blue line shows optimized trajectory

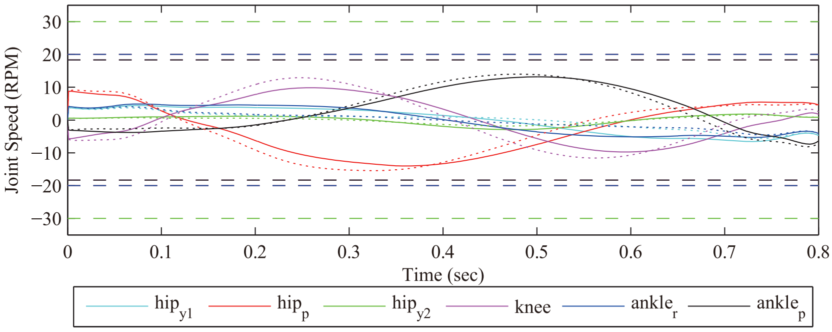


Fig. 6. Swing Foot Joint Speed Trajectory for Fixed Body Angle Case. Dotted lines show original joint speed trajectory. Solid line shows the joint speed trajectory after the optimization. Horizontal dashed lines show joint speed limits of the corresponding colored joints.

were used as parametric curves and the boundary conditions were optimized with nonlinear optimization solver. Also to generate near-optimal and feasible trajectory, online trajectory regeneration method was presented that deforms original trajectory with new boundary conditions with null-space projection method. The optimization method showed reduction in maximum joint speed by 6.8% and 18% for the case of fized and moving body yaw angle respectively. The hardware test showed 0.35 m stride stably for both fixed body angle case and variable yaw angle case, which is 40% improvement on the maximum possible stride compared to original heuristic approach. The experiment showed efficacy of the proposed method.

Acknowledgment

The first author would like to thank "Study Program at the Overseas Universities" by Graduate School of Information Science and Technology, The University of Tokyo, for the travel expenses to Germany. The second and the third author acknowledge support by the Initiative and Networking Fund of the Helmholtz Association (VH-NG-808).

References

[1] M. Vukobratovic and J. Stepanenko, "On the stability of anthropomorphic systems," *Mathematical Biosciences*, vol. 15, pp. 1–37, 1972.

[2] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," in *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2001, pp. 239–264.

[3] T. Sugihara and Y. Nakamura, "Whole-body Cooperative Reaction Force Manipulation on Legged Robots with COG Jacobian involving Implicit Representation of Unactuated Coordinates," *Journal of Robotics Society of Japan*, vol. 24, no. 2, pp. 222–231, 2006, in Japanese.

[4] C. Ott, C. Baumgartner, J. Mayr, M. Fuchs, R. Burger, D. Lee, O. Eiberger, A. Albu-Schäffera, M. Grebenstein, and G. Hirzinger, "Development of a Biped Robot with Torque Controlled Joints," in *Proc. of 10th IEEE-RAS Int'l Conf. on Humanoid Robots*, 2010, pp. 167–173.

[5] C. Ott, O. Eiberger, J. Englsberger, M. A. Roa, and A. Albu-Schäffer, "Hardware and Control Concept for an Experimental Bipedal Robot with Joint Torque Sensors," *Journal of Robotics Society of Japan*, vol. 30, no. 4, pp. 378–382, 2012.

[6] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based control of robot manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[7] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery," in *Proc. of 6th IEEE-RAS Int'l Conf. on Humanoid Robots*, 2006, pp. 200–207.

[8] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal Walking Control based on Capture Point Dynamics," in *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2011, pp. 4420–4427.

[9] MathWorks, "Mathworks home page," http://www.mathworks.com.

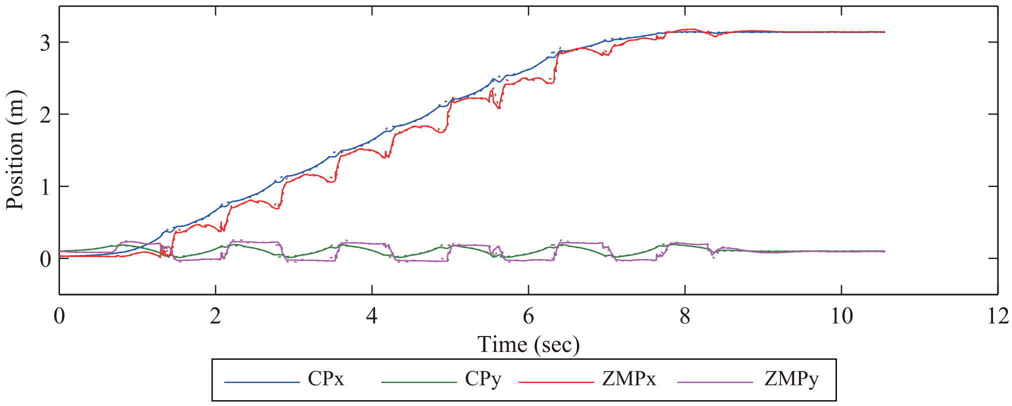


Fig. 7. Capture Point (CP) and ZMP Trajectory of the Case with Body Yaw Angle Fixed. Solid line shows actual value and the dotted line shows desired value.

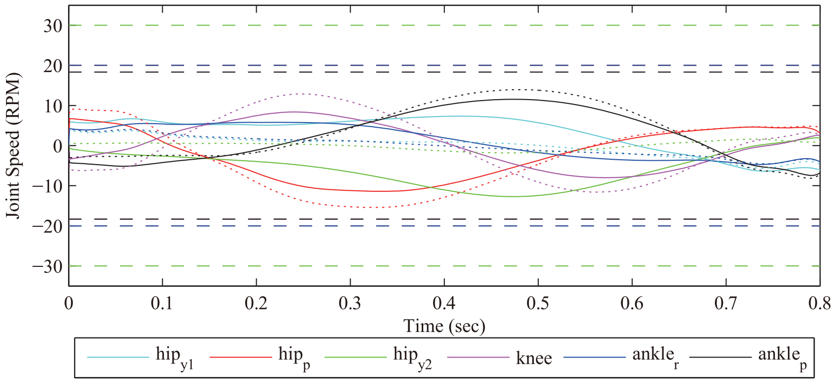


Fig. 8. Swing Foot Joint Speed Trajectory for Variable Yaw Body Angle Case. Dotted lines show original joint speed trajectory. Solid line shows the joint speed trajectory after the optimization. Horizontal dashed lines show joint speed limits of the corresponding colored joints.

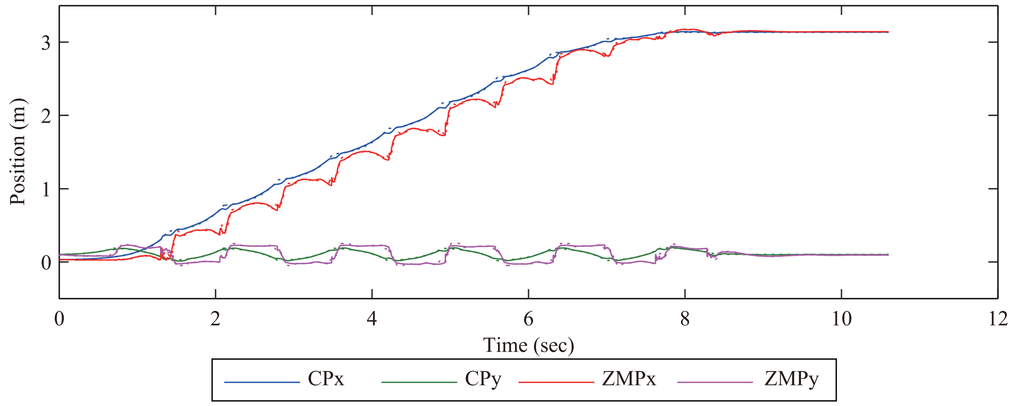


Fig. 9. Capture Point (CP) and ZMP Trajectory of the Case with Body Yaw Movement. Solid line shows actual value and the dotted line shows desired value.