We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

11

Real-Time Optimization Approach for Mobile Robot

Hiroki Takeuchi
University of Michigan
FXB Building 1320 Beal Avenue Ann Arbor, MI 48109-214
U.S.A.

1. Introduction

Optimization technique has been applied at many engineering aspects. Especially, it was used as a trajectory analysis of aircraft or spacecraft at the beginning. As computer environment grows up, such application comes to focus on more complicated and dynamic problem. One of them is mobile robot.

On another front, optimization algorithm itself has been developed for on-line use. Classic method like as gradient method has been applied for off-line computing. Off-line technique can not be applied to real-time use like as robot control. However, after new method such as receding horizon control was emerged, the ability for real-time control use comes to be practical.

What the brain of human or animal executes is, as it were, "optimization". As the result, it generates very smooth motion. Our goal is to realize such optimization and make mobile robot motion more smart and intelligent.

Receding Horizon Control is one of potential optimization technique. This algorithm is oriented for on-line use and practical optimization. This chapter provides some results of, (1) RHC formulation including ZMP balance condition, (2) RHC formulation including ZMP balance condition and Swing leg state variable constraint. Generally, theoretical treatment of Zero Moment Point is hard to solve. ZMP variable has an aspect of intermediate variable and it is somewhat mysterious. The technique can give a deal for such formulation. The numerical model is defined as simple one in the formulation of (1), then the proposed method can be applied to any legged robot which has to handle ZMP variable. Simulation results are also provided.

2. Real-Time Optimization

Generally, Gradient Method is most popular method in optimization technique. However, its long calculation time has made hurdle to use at real-time optimization. The procedure which arbitrary trajectory for whole time converges into optimal solution using gradient makes a defect. This procedure also may cause the trajectory sink to local minimum.

There are several real-time optimization algorithms called Receding Horizon Control. Those have good and bad points. In this chapter, two methods are introduced. The one is Receding Horizon Gradient Method, the other is backward sweep method which use transversality condition and it is used for the application in section 3 and section 4.

Source: Mobile Robots, Moving Intelligence, ISBN: 3-86611-284-X, Edited by Jonas Buchli, pp. 576, ARS/plV, Germany, December 2006

2.1 Receding Horizon Gradient Method

2.1.1 Continuation Method

Gradient Method has conventionally eliminated a problem that the initial condition of state equation and terminal condition of co-state equation are known although terminal condition of co-state equation and initial condition of co-state equation is unknown in TPBVP. Gradient Method is the method that the initial trajectory which is assigned as whole time (t=0 -> t=t_f) on real-time axis converges to optimal solution along its gradient. Defects of this method are;

(1) It could converge to minimum solution
(2) It takes a good amount of time to converge.
(3) This initial trajectory must be considered for immediate convergence each time.

Then a Continuation Method eliminated these problems in place of Gradient Method (Ohtsuka,1997). If the interval time in Euler-Lagrange equations set 0, the solution comes to be trivial. The optimal solution can be chased to extend the time little by little based on this trivial solution. Generally, Predictor-Corrector Method is used for pursuit in Continuation Method. However it is too slow to execute on real-time control sequence.

Then a method which transverse condition is applied to balance to 0 eliminated this problem (ohtsuka,1997). The process to handle matrix operation needs longer calculation time if the numerical model is larger scale. In this research, a proposed method eliminates formula operation without Euler-Lagrange equations as much as possible. Such attempt may be also said that it returns to its basic focus on.

2.1.2 Gradient

When the terminal time T on performance index interval perturbs T+dT, the trajectory also perturbs. The extended performance index is described as:

$$J^* = \varphi[x^*(t,\tau)] + \int_t^{t+T} L + \lambda^{*T}(t,\tau)\cdot\{f[x(t,\tau),u(t,\tau)] - \dot{x}(t,\tau)\}d\tau \tag{2.1}$$

Then the perturbation is described as:

$$\delta J^* = [\frac{\partial\varphi}{\partial\tau} + H]_{\tau=T}dT + [\varphi_x[x(t,\tau)] - \lambda^*(t,\tau)]_{\tau=T}dx(t,T) + \int_t^{t+T}[\{H_x + \dot{\lambda}(t,\tau)\}\cdot\delta x + H_u\delta u + \delta\lambda^{*T}(f-\dot{x})]d\tau \tag{2.2}$$

When the terminal time T on performance index interval perturbs T+dT, it causes $\delta x, \delta\lambda, \delta u$. δu dominates $\delta x, \delta\lambda$ by state equation and co-state equation.

If the trajectory satisfies Euler-Lagrange equations, the gradient must be 0. If the trajectory perturb from the optimal solution, it cause the generation of gradient. In Continuation Method, the gradient of the initial trajectory equals 0, then it perturbs T+dT, causes a bit of gradient, and it is recovered.

2.1.3 Sampling Interval

The movement of performance index interval along real-time is executed each sampling interval. Although the time length of the performance index interval is extended, it is little and changes smoothly. This enables the time length re-scaled along the extension.

In reference (Ohtsuka, 1997), the number of the sampling interval arrays is fixed. In this case the sampling interval is growing longer along real-time, and it must be considered how to get re-scaled initial state value, co-state value, and input value. In this paper, the time length of the array is fixed; the number of these arrays is growing along real-time. $d\tau = cons\tan t = dt$. This aims for practical use. It can replace the initial array of x, λ, u on performance index interval as the next step array of x, λ, u on real-time directly. The simulation results confirm that this is feasible.
Therefore the algorithm proposed here is:

(1) t=0, $\tau = 0$, trivial solution
(2) $T_{i+1} = T_i + d\tau$
(3) Rescale trajectory array of input variable on τ axis
(4) Calculate trajectory of state variable with state equation
(5) Calculate trajectory of co-state variable with co-state equation
(6) Calculate the gradient H_u
(7) If the gradient sufficiently closed to 0, then go to (10)
(8) Update the trajectory of input variable on τ axis $u_{new} = u_{old} + \alpha \cdot gradient$
(9) $\alpha = \alpha + step$; Go to (4)
(10) The initial array of input variable on τ axis replaces the next step array of input variable on t axis.
(11) Calculate state value and co-state value from (10)

Various rescaling could be considered at (3). One of them is to use the input variable trajectory one step ago and rescale it like as Fig**.** 2. Somewhat gradient is invoked by rescaled input variable trajectory, however, such gradient could be expected sufficiently small from the viewpoint of Continuation Method.

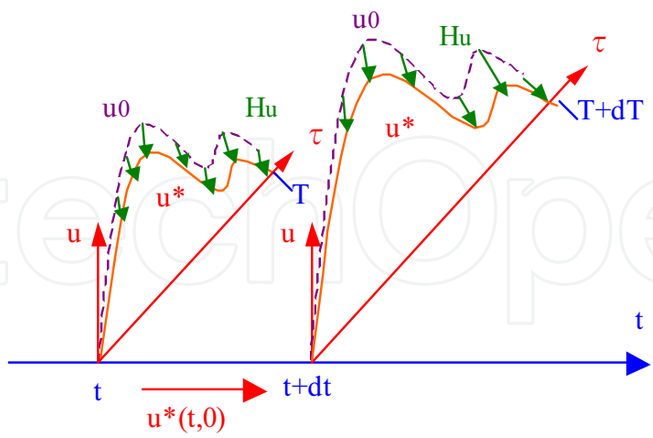


Fig. 1. Differential changes in the terminal time.

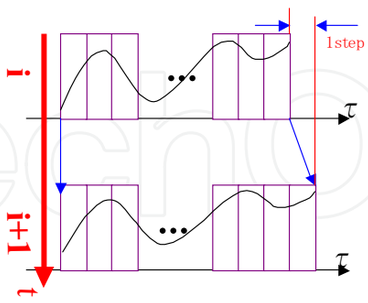


Fig. 2. Scaling for input variable.

A simple example is arranged here. The state equation is described as:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} (1-x_1^2(t)-x_2^2(t))x_1(t)-x_2(t)+u(t) \\ x_1(t) \end{bmatrix} \tag{2.3}$$

The state variables are $x_1(t), x_2(t)$, and input variable is $u(t)$. Performance index is described as:

$$J = 2(x_1^2(t_f)+x_2^2(t_f))+\int_0^{t_f}(x_1^2(t)+x_2^2(t)+u^2(t))dt \tag{2.4}$$

The evaluated interval is moving and extended along real-time in Receding Horizon Control procedure, then it is described as:

$$J = 2(x_1^2(t,T)+x_2^2(t,T))+\int_t^{t+T}(x_1^2(t,\tau)+x_2^2(t,\tau)+u^2(t,\tau))d\tau \tag{2.5}$$

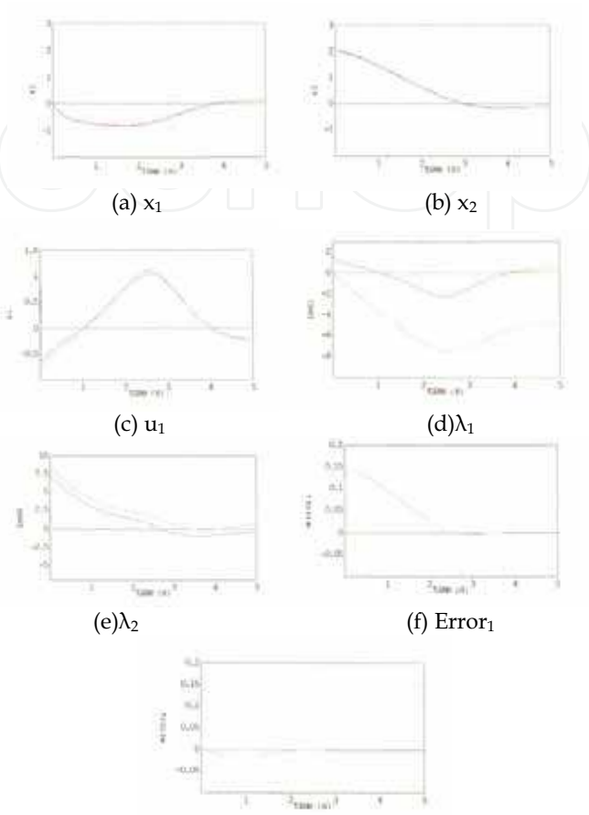
Fig. 3 shows the comparison between conventional gradient method (Fletcher-Reeves Method) and proposed method. The trajectories of state variables and control inputs are matched each other. Fig. 3 (f), (g) error values transition described as Equation (2.6) endorse this fact. The each error is closed to 0 sufficiently.

$$error(t) = \lambda^*(t,T)-\varphi[x^*(t,T)] \tag{2.6}$$

Mathematica3.0 on Windows OS executes this calculation.

	RHGM	**Gradient Method**
Simulation Time	5.0 s	5.0 s
dt	10.0 ms	50.0ms
Continuation Terminal Time	$T_{i+1}=T_i+10.0$ms	-
Maximum of Terminal Time	0.5 s	-
Maximum number of iteration in a step	40	30
Initial Condition x	{0.0, 2.0}	{0.0, 2.0}
Reference x_f	{0.0, 0.0}	{0.0, 0.0}

Table 1. Simulation Data.



(a) x_1 (b) x_2

(c) u_1 (d)λ_1

(e)λ_2 (f) $Error_1$

(g) $Error_2$

Fig. 3. Example (Dashed line: gradient method, solid line: RHGM).

A. Nonlinear link system

Nonlinear three-link system can also be solved by proposed method. The state equation is described as:

$$\begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \\ \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ \dot{\theta}_3(t) \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ \dot{\theta}_3(t) \\ M^{-1}(\Theta) \cdot (u(t) - V(\Theta, \dot{\Theta}) - G(\Theta)) \end{bmatrix} \tag{2.7}$$

$$\Theta = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \theta_3(t) \end{bmatrix}$$

The equation involves nonlinearity of vertical three-link system. M means inertial matrix, V means carioles term, G means gravity term. Performance index is defined as:

$$J = (x_f(t,T) - x(t,T)) \cdot S_f \cdot (x_f(t,T) - x(t,T)) + \int_{\tau}^{\tau+T} (u \cdot R \cdot u + x \cdot Q \cdot x) d\tau \tag{2.8}$$

Fig. 5 shows the result of this simulation. Table 2 describes calculation parameters and physical parameters of the links. Fig. 5 (q)-(v) shows the error of the transverse condition (Equation(2.6)). Simulation was done on Mathematica 5.0. One of the advantages of RHC is that it can treat with singular target point. RHC algorithm does not contain Jacobian matrix, then it can treat with such problem. The simulation result shows that the three link manipulator reaches to the singular target attitude.

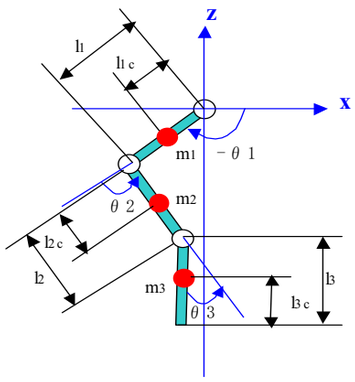
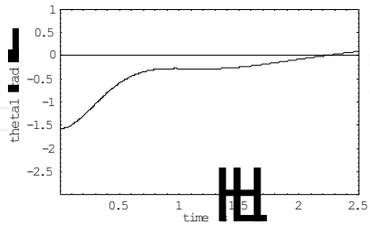


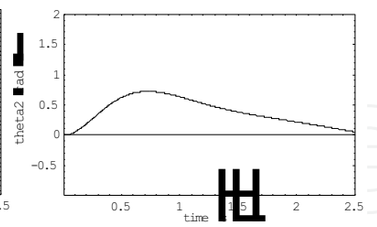
Fig. 4. Nonlinear link system.

Simulation Time	2.5 s
dt	5.0 ms
Continuation Terminal Time	$T_{i+1}=T_i+dt$
Maximum of Terminal Time	0.5 s
Maximum number of iteration in a step	40
Initial Condition	{-1.57,0.0,0.0,0.0,0.0,0.0}
Reference Condition	{0.0,0.0,0.0,0.0,0.0,0.0}
S_f	{1.0,1.0,1.0,0.1,0.1,0.1}
R	{1.0,1.0,1.0}
Q	{1.0,1.0,1.0,1.0,1.0,1.0}
m_1	0.5kg
m_2	0.5kg
m_3	0.5kg
l_1	0.3m
l_{1c}	0.2m
l_2	0.3m
l_{2c}	0.2m
l_3	0.3m
l_{3c}	0.2m

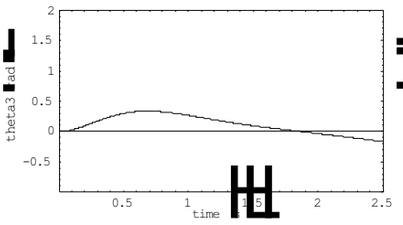
Table 2. Simulation Data



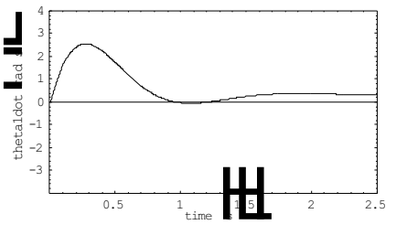
(a) θ_1



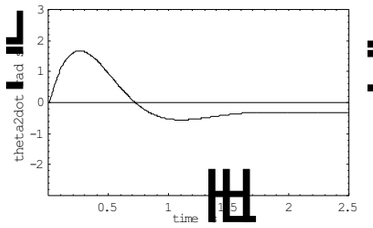
(b) θ_2



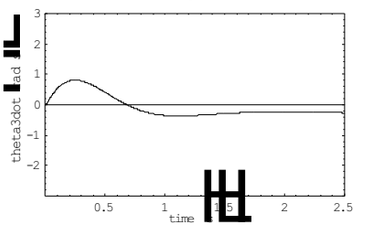
(c) θ_3



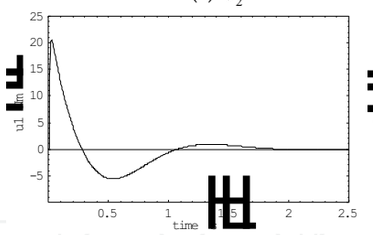
(d) $\dot{\theta}_1$



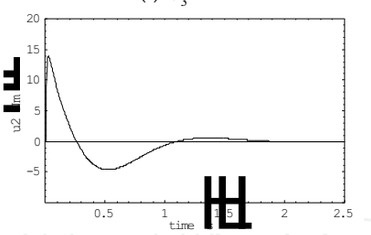
(e) $\dot{\theta}_2$



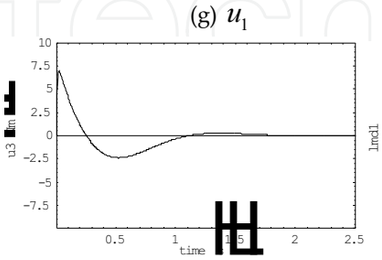
(f) $\dot{\theta}_3$



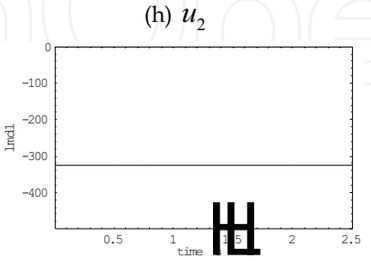
(g) u_1



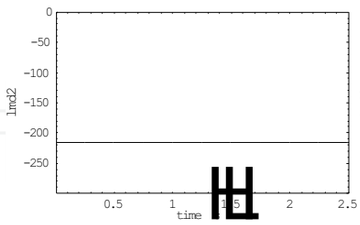
(h) u_2



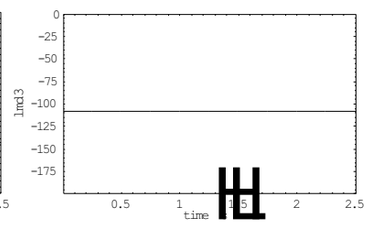
(i) u_3



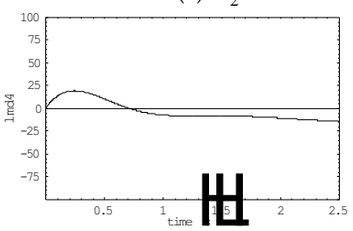
(j) λ_1



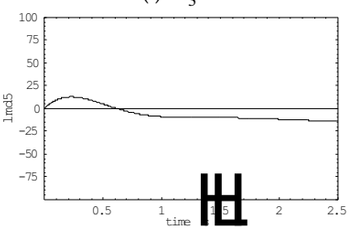
(k) λ_2



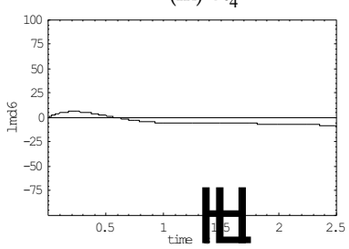
(l) λ_3



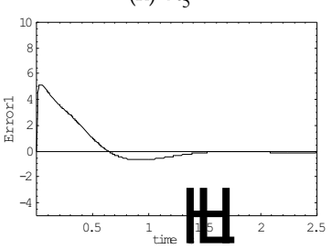
(m) λ_4



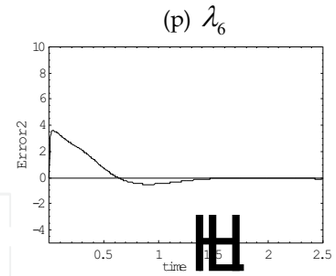
(n) λ_5



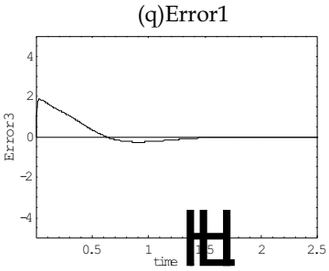
(p) λ_6



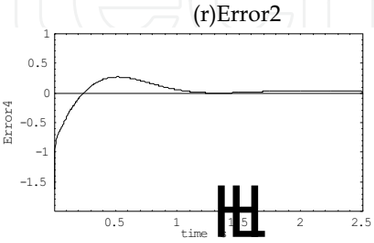
(q)Error1



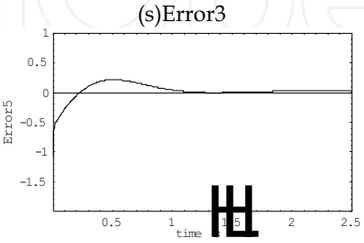
(r)Error2



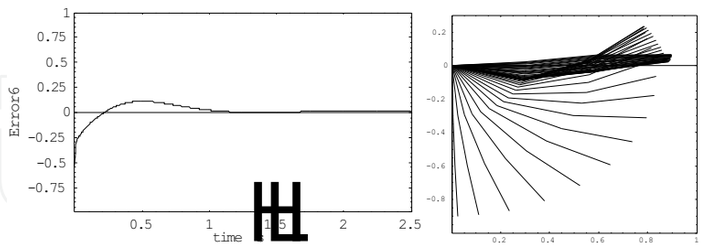
(s)Error3



(t)Error4



(u)Error5



(v)Error6 (w)Stick Figure

Fig. 5. Nonlinear 2- link system.

3. Particular Condition for Legged Robot – ZMP Balance Condition –

Legged robot has specific balance condition attributable to the unstable dynamics. It needs to contrive ways to involve such condition into formulation.

While the legged robot is standing on one foot, the point at which the center of gravity **(**C.G.**)** of the robot is projected onto the ground must be located on the sole plane to enable it static walk. While standing on two feet, there must be a point on the plane, which connects both the soles. While standing on four feet, there must be a point on the polygon, which consist of the four soles. While the robot moves, in order to be stabilized dynamically and to walk, the same concept is required. Generally, this is called ZMP **(**Zero Moment Point, **(**Vukobratovic,1969**))**. ZMP within sagittal plane can be expressed as follows from link i=0 to i=n.

ZMP is the point on the ground where ground reaction forces are applied.

$$x_{zmp} = \frac{-\sum_{i=0}^{n} m_i(-g+\ddot{z}_i)x + \sum_{i=0}^{n} m_i \ddot{x}_i z_i}{\sum_{i=0}^{n} m_i(-g+\ddot{z}_i)} \tag{3.1}$$

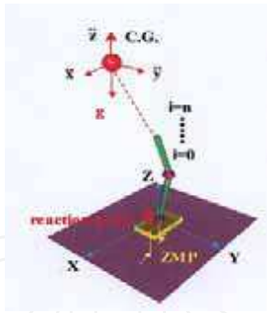


Fig. 6. Definition of Zero Moment Point.

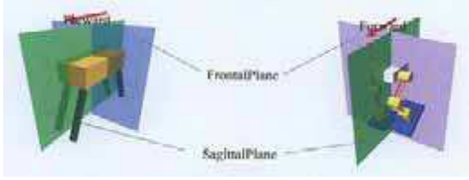


Fig. 7. Definition of sagittal plane and frontal plane.

g is gravity acceleration and m is the mass of each link. If the "M" is represented for the whole mass,

$$x_{zmp} = \frac{-M(-g+\ddot{z})x + M\ddot{x}z}{M(-g+\ddot{z})} \tag{3.2}$$

This equation means that the sum total of moment of the point-mass around the origin of the coordinate balances with the moment generated by the ZMP distance from the origin and the reaction force from the ground. If the ZMP is located in the polygon constituted by the soles as well as the point at which the C.G. projects itself onto the ground in a static walk, the robot is stabilized and a dynamic walk can be carried out. If the ZMP runs-over from this polygon, it will cause the robot to fall and it cannot continue to walk.
An attempt to converge the ZMP to a referenced ZMP trajectory by using feedback control in recent years has been performed **(**Hirai, K., Hirose, M., Haikawa,Y., Takenaka, T., 1998). Then, how a ZMP reference trajectory could be generated poses the next problem.
In the conventional research of legged robot, there are two variables "x" and "z" in Equation(3.2) and poses a problem in solving the ZMP variable. Because it is not solved uniquely. An optimization problem must be solved to obtain a solution. If the condition , which holds a center of gravity position at fixed height, is added, we can avoid this problem temporarily. Then, the variable $\ddot{z}$ in the equation is set to 0, the equation could be described as follows, and a pseudo solution is uniquely obtained.

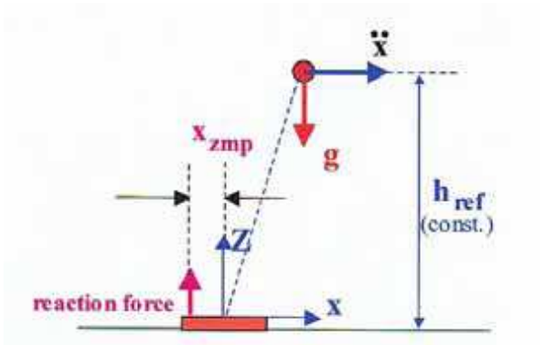


Fig. 8. Pseudo ZMP.

In the conventional legged robot research, one of big problems was to compute ZMP by Equation(3.2) since there are two variables of "x" and "z". To avoid this problem, some treatments had been concerned(mitobe,2002)}. The main concept was to make the numerical model unique. Nonlinear dynamic equation is linearized adding constraints and reduced to unique equation like as A constraint below is added:

$$z = k \cdot x + h_{ref} \tag{3.3}$$

h_{ref} denotes a fixed height. Then we have linear equation:

$$\ddot{x} = \frac{g}{h_{ref}} \cdot x + \cdot \frac{1}{m \cdot h_{ref}} \tau \tag{3.4}$$

τ denotes ankle joint torque. The value of ZMP can be obtained from τ and sensed value of floor reaction force. Then the ZMP could be in proportion with the acceleration of x. One of

the defects in this equation is that the necessary torque for whole the robot is collected on the ankle joint. Redundancy, which the robot possesses, could not be utilized effectively. Equation(3.3) is defined arbitrarily by designer.

(1) Excessive torque for whole the robot is converged to the ankle joint
(2) Robot motion is restricted because the constraint is adopted (The motion is on a linear line)
(3) Solutions for another joints without the ankle joint could not be obtained

The second item implies that robot motion is not natural. What it takes to utilize redundancy of a robot is optimization. Since iterative calculation is needed in the optimization by the gradient method, such technique usually turns into off-line calculation. However, in a robot control that the real-time performance is required, off-line optimization is disadvantageous. When an unexpected situation appears, it could not be coped with.

Many engineering applications require real-time solutions of optimization problems. However, traditional algorithms for digital computers may not provide real-time optimization. An attractive and promising approach was introduced to real-time solutions for optimization problems known as Receding Horizon Control. This new optimization technique goes into the practical usage stage. Since RHC does not use a gradient method for optimization, it can carry out calculation processing of the optimal solution in short time such as a real-time control interval. Although much research has been conducted in respect to the theory, applying RHC to robotics still has no actual example. This paper describes ZMP control of the legged robot using RHC proving that real-time optimization is available. Furthermore, it proposes a method of generating the optimum ZMP reference (Takeuchi,2003).

3.1 Equality Constraint Formulation

RHC formulation without constraints has been performed backwards. In this section, RHC containing the equality constraint is focused on and explained.

The state equation to treat,

$$\dot{x}(t) = f[x(t), u(t)] \tag{3.5}$$

As equality constraint,

$$C[x(t), u(t)] = 0 \tag{3.6}$$

If equality constraint condition can be used, it is convenient when formulizing a problem like real-time control of a robot.

The performance index is defined as,

$$J = \phi[x^*(t+T)] + \int^{+T} L[x^*(t,\tau), u^*(t,\tau)]d\tau \tag{3.7}$$

RHC has added superscript * to the variable on the time-axis τ which moves, in order that the evaluation section may move with time. The left side of the bracket (t,τ) means the real time "t", and the right side of this bracket means the time on the τ axis. Hamiltonian is described as,

$$H = L + \lambda^{*T} \cdot f + \rho^{*T} \cdot C \tag{3.8}$$

λ, ρ are co-state variables. Euler-Lagrange equations are described as,

$$\dot{x}^*(t,\tau) = H_\lambda^T \tag{3.9}$$

$$\dot{\lambda}^*(t,\tau) = -H_x^T \tag{3.10}$$

$$H_u = 0 \tag{3.11}$$

$$C[x^*(t,\tau), u^*(t,\tau)] = 0 \tag{3.12}$$

$$x^*(t,0) = x(t) \tag{3.13}$$

$$\lambda^*(t,T) = \phi_x^T[x^*(t,T)] \tag{3.14}$$

It considers obtaining a solution using the continuation method (Ohtsuka, 1997). The perturbation from an optimal path is described as:

$$\delta\dot{x} = f_x^T \cdot \delta x - f_u^T \cdot \delta u - f_\rho^T \cdot \delta\rho \tag{3.15}$$

$$\delta\dot{\lambda} = -H_{xx} \cdot \delta x - f_x^T \cdot \delta\lambda - H_{xu} \cdot \delta u - H_{x\rho} \cdot \delta\rho \tag{3.16}$$

$$H_{xu} \cdot \delta x + f_u^T \cdot \delta\lambda + H_{uu} \cdot \delta u - H_{u\rho} \cdot \delta\rho = 0 \tag{3.17}$$

$$C_x \cdot \delta x + C_u \cdot \delta u = 0 \tag{3.18}$$

Here, δu and $\delta\rho$ are eliminable if Equation**(3.17)**,Equation**(3.18)** are solved as simultaneous equations. Then Equation**(3.15)**,Equation**(3.16)** are described as:

$$\frac{d}{dt}\begin{bmatrix}\delta x \\ \delta\lambda\end{bmatrix} = \begin{bmatrix}A & B \\ C & D\end{bmatrix}\begin{bmatrix}\delta x \\ \delta\lambda\end{bmatrix} \tag{3.19}$$

$$A = f_x + f_\rho \cdot H_{u\rho}^{-1} \cdot H_{ux} - f_\rho \cdot H_{u\rho}^{-1} \cdot H_{uu} \cdot C_u^{-1} \cdot C_x - f_u \cdot C_u^{-1} \cdot C_x \tag{3.20}$$

$$B = f_\rho \cdot H_{u\rho}^{-1} \cdot f_u^T \tag{3.21}$$

$$C = -H_{xx} + H_{xu} \cdot C_u^{-1} \cdot C_x - H_{x\rho} \cdot H_{u\rho}^{-1} \cdot H_{ux} - H_{x\rho} \cdot H_{u\rho}^{-1} \cdot H_{uu} \cdot C_u^{-1} \cdot C_x \tag{3.22}$$

$$D = -f_x^T - H_{x\rho} \cdot H_{u\rho}^{-1} \cdot f_u^T \tag{3.23}$$

The subsequent calculation method follows the continuation method which Ohtsuka and Fujii developed(Ohtsuka,1997). This is explained briefly below. This technique pursues the optimal solution so that the error F of the transversality conditions of an Euler-Lagrange equation is converged to 0.

$$\frac{d}{dt}F[\lambda(t), x(t), T(t)] = Coeff \cdot F[\lambda(t), x(t), T(t)] \tag{3.24}$$

Thus, F can be stabilized. The equation below is assumed here.

$$\delta\lambda^*(t,\tau) = \delta S^*(t,\tau) \cdot \delta x^*(t,\tau) + c^*(t,\tau)dt \tag{3.25}$$

This is substituted for Equation(3.19).

$$S_\tau^* = D \cdot S^* - S^* \cdot A - S^* \cdot B \cdot S^* + C \tag{3.26}$$

$$c_\tau^* = (D - S^* \cdot B) \cdot c^* \tag{3.27}$$

This terminal value is acquired from Equation**(3.25)**. An $S^*(t,0), c^*(t,0)$ will be acquired if it finds the integral from the terminal value along time reversely.

$$\dot{\lambda}(t) = S^*(t,0) \cdot \dot{x}(t) + c^*(t,0) \tag{3.28}$$

Then, optimized $\lambda(t)$ will be obtained if it integrates with the upper equation on real time. Also optimized u**(**t**)** can be obtained from $H_u = 0$.

3.2 Model Expression as a Point Mass

Modeling of robot mechanics has been studied for many years. One of the ways is to describe nonlinearity of the robot precisely. However such dynamic equations leads to a result that the equation itself is too much complicated to treat. Such modeling also could not be applicable to even similar type robots. From this point of view, a concept of simple modeling has been emerged. Modeling simply leads to be basic and flexible to apply it to various type controlled objects. The fundamental treatment should be formulated at first, and then applied to more complicated modeling properly.

However, overdo of reduction and linearization in modeling leads to many defects mentioned in 5.1 Introduction. The modeling must be simple, but also, must be with wide application.

In this study, when a legged robot is modeled, the whole robot is treated as a point mass of most fundamental case. Treating the whole robot center of gravity as inverted pendulum is a technique generally performed. According to such simple modeling, a method applicable to a biped robot, a quadruped robot, and other multi-legged type robots can be proposed so that Chapter**\ref{**chap:application**}** may describe. First, in order to help understanding, it deals with a problem at a 2-dimensional plane. As an input of a system, it sets setting up the acceleration of axis "x" and "z", $u_x = \ddot{x}$, $u_z = \ddot{z}$. Gravity is applied to a perpendicular lower part.

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ z(t) \\ \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \\ u_x(t) \\ u_x(t) - g \end{bmatrix} \tag{3.29}$$

Although this is considered on Sagittal plane, if it considers at Frontal plane,

$$\frac{d}{dt}\begin{bmatrix} y(t) \\ z(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} \dot{y}(t) \\ \dot{z}(t) \\ u_y(t) \\ u_y(t) - g \end{bmatrix} \tag{3.30}$$

This modeling does not include ankle joint torque explicitly. Then the problem mentioned in 5.1 Introduction could not be emerged. The potential of the redundancy of

the robot could be left and it produces extensive utility. Furthermore, we can obtain ZMP trajectory.

3.3 Equal Constraints

Difficulties in taking the ZMP variable into a state equation as a state variable complicates the problem in formulation. Because the right side of Equation**(3.2)** has the dimension of acceleration, it causes differentiation of acceleration. Then, the use of the equality constraint expressing ZMP eliminates this problem.

$$x_{zmp}(t)\cdot(u_z(t)-g)=x(t)\cdot(u_z(t)-g)-z(t)\cdot u_x(t) \tag{3.31}$$

This means that the total moment by the acceleration inputs and gravity balances with the moment by the ZMP distance and reaction force from the ground. ZMP is the point of satisfying Equation**(3.31)**. If ZMP is located in the polygon constituted by the sole plane, it can support the reaction force without generating any moment. Furthermore, this paper proposes that the 3rd input u_{zmp} substitute for x_{zmp}, then this idea compliments using the ZMP variable in formulation.

It is not necessary for u_{zmp} to be included in the state equation.

$$u_{zmp}(t)\cdot(u_z(t)-g)=x(t)\cdot(u_z(t)-g)-z(t)\cdot u_x(t) \tag{3.32}$$

If ZMP is treated as one of the inputs, when an optimum solution is calculated, the optimum ZMP input will be obtained simultaneously. At this point, the equal constraint has an important role.

The performance index is created using norm of inputs. Here, features include that the term of u_{zmp} is added in the performance index. Thus, by setting up the solution , which minimizes the norm of each axial acceleration and the ZMP sway, will be calculated. Considering that the ZMP may not sway within the sole of the robot, this has lead to the design of the robot's sole to be as small as possible.

$$J=X^T\cdot S_f\cdot X+\int^{+T}(X^T\cdot Q\cdot X+U^T\cdot R\cdot U)d\tau$$

$$X=\left[x_f-x(t,T),y_f-y(t,T),\dot{x}_f-\dot{x}(t,T),\dot{y}_f-\dot{y}(t,T)\right]^T \tag{3.33}$$

$$U=\left[u_x^*(t,\tau),u_z^*(t,\tau),u_{zmp}^*(t,\tau)\right]^T$$

S_f and R are diagonal weight matrix.

3.4 Application

A legged robot is generally a nonlinear mechanical multi-link system. In order to compensate such nonlinearity, using this technique together with the conventional nonlinear control technique is also worth consideration. The real-time optimum solution using RHC can be used as reference trajectories. On the other hand, nonlinear control in modern control theory performs control of the whole multi-link system of the robot to converge to the reference trajectories.

When we have a powerful CPU for real-time control, it is possible that it formulates everything in a nonlinear model using RHC alone.

Using the conventional control method for the legged robot, the optimum reference trajectory have to be prepared before the real-time control. If the robot encounters an unexpected situation, the robot cannot cope with it. In regards to this point, the technique proposed is practical.
The optimum ZMP input, which is obtained using RHC, can generate the reference trajectory. Then a control system block diagram could be constructed so that actual ZMP may follow this reference trajectory. Since u_x, u_y, and u_z are obtained for optimum inputs, they are newly regarded as reference trajectories.
Thus, this simple method is applicable also to a biped robot, a quadruped robot, and other multiped robots.

3.4.1 In the Case of a Biped

In the case of the biped, the phase of leg behavior could be divided into a support phase and a swing phase. The ZMP must be respectively settled in the sole planes. The ZMP must transition in these planes, if the initial point and the terminal point are set as in Fig.**9**. Control of the nonlinear force is performed in the nonlinear control part as is shown in the Fig.**10** control block diagram. Collecting the calculation results in which each parameter is changed makes it possible to predict the range of ZMP sway in a sole. This is a very interesting point. In design of a biped robot, an index can be obtained which has an influence on the design dimensions of the sole plane.

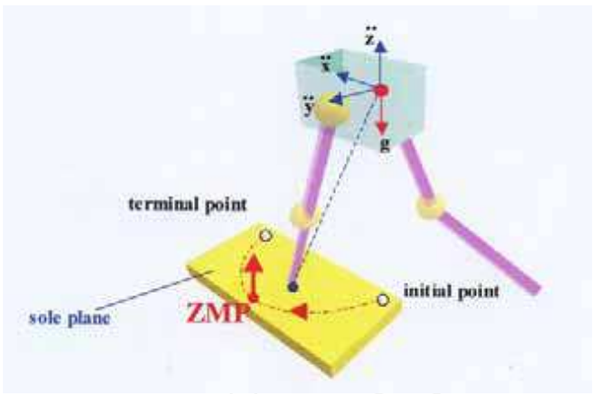


Fig. 9. Case of Biped.

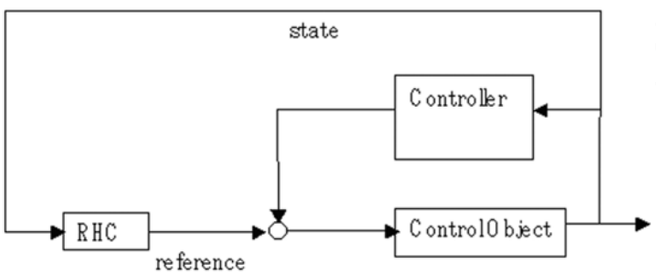


Fig. 10. Control Block Diagram.

3.4.2 In the Case of a Quadruped

The legs can be treated as the support legs pair and the swing legs pair by turns in ideal gaits. The polygon which is constituted by some sets of legs can secure a larger plane compared with a biped robot's case. For example, when a quadruped robot performs a "trot gait", it is necessary to repeat a support phase – swing phase for the diagonal leg entirely by turns. In Fig.**11** , step planes cross in the trot gait. If the starting point of ZMP is set at the tip of this plane, it can run to the tip of the following plane. Then a stable locomotion pattern will be achieved.

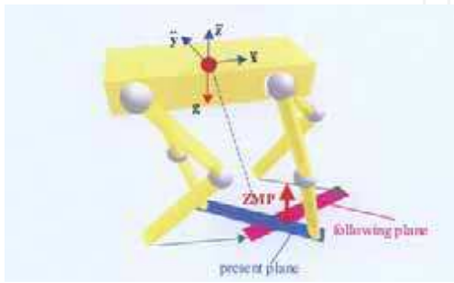


Fig. 11. Case of Quadruped.

4. Swing Leg Condition – State Variable Constraints –

The point mass modeling for formulation of Receding Horizon Control is introduced in the preceding section. That idea is to look at the robot model in perspective. We are in the next stage how the swing legs should be treated. The legged robot motion can be categorized into two phases support leg phase and swing leg phase depending on whether or not they are in contact with the floor while the legged robot is moving. The support legs support the robot against gravity while the swing legs are swung forward and then prepared for the subsequent support phase.

Since a real robot's legs behave nonlinearly, nonlinear forces interfere with the motion of the swing legs when they move. It is not practical to include all nonlinearities of the robot in a state equation. Therefore, this chapter proposes a simple formulation that reflects the nonlinearity of the motion of swing legs as far as possible. The root joint of the swing leg is connected to the center of gravity of the robot in sagittal plane and the acceleration is transferred to the root joint while the center of gravity satisfies the conditions of the ZMP constraint. This formulation can be easily applied to multi-legged robots such as biped, quadruped, and hexapod by simple addition of the equations of motion of the swing legs.

It is necessary to apply the constraint on the swing legs such that their position is always above floor level. The absence of this constraint could create a situation in which the legs would be positioned below the floor. Constraints such as these can be described as inequality state variable constraints. This section describes a method for solving the problem by means of the slack variable method.

4.1 Modeling of Swing Leg

Generally, a mathematical model of a legged robot increases in complexity if we try to describe the nonlinearity of its movement precisely, with the result that the equations of motion become

too complex for the robot control to apply. A mathematical model that requires less calculation is needed to shorten the control interval and allow real-time control of the swing legs.
(Takeuchi,2003) describes the ZMP condition on the sagittal plane by using a constraint equation.

$$x_{zmp}(t)\cdot(u_z-g)=x(t)\cdot(u_z(t)-g)-z(t)\cdot u_x(t) \tag{4.1}$$

This indicates that the moment, which consists of x_{zmp} and the reaction force of the floor, balances with the gross moment around the origin of the coordinate system. Kinetic balance is maintained as long as x_{zmp} is in the area of the sole **(** biped **)**, or in the area of the polygon described by the points where the toes of the grounding legs touch the floor **(**quadruped**)**.
The state equation of the center of gravity of a robot in sagittal plane is described as follows.

$$\frac{d}{dt}\begin{bmatrix}x(t)\\ z(t)\\ \dot{x}(t)\\ \dot{z}(t)\end{bmatrix}=\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\\ u_x(t)\\ u_z(t)-g\end{bmatrix} \tag{4.2}$$

At practical control of legged robot, the control of swing legs must also be taken into account. We have therefore described the swing legs as a nonlinear two-link coordinate system and appended it to the state equation of the center of gravity of the robot. This approach is based on the assumption that the root joint of the swing leg will receive the acceleration of the center of gravity of the whole robot.

$$\frac{d}{dt}\begin{bmatrix}x(t)\\ z(t)\\ \theta_1(t)\\ \theta_2(t)\\ \dot{x}(t)\\ \dot{z}(t)\\ \dot{\theta}_1(t)\\ \dot{\theta}_2(t)\end{bmatrix}=\begin{bmatrix}\dot{x}(t)\\ \dot{z}(t)\\ \dot{\theta}_1(t)\\ \dot{\theta}_2(t)\\ u_x(t)\\ u_z(t)-g\\ M^{-1}(\Theta)(u(t)-V(\Theta,\dot{\Theta})-G(\Theta))\end{bmatrix} \tag{4.3}$$

$$\Theta=\left[\theta_1(t),\theta_2(t)\right]^T$$

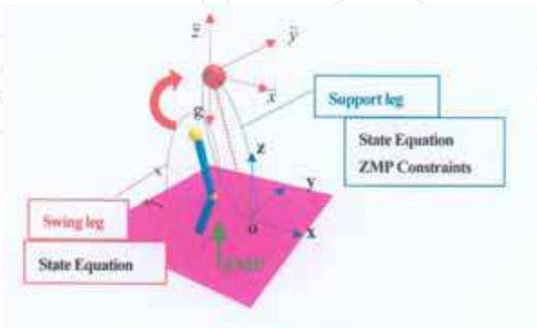


Fig. 12. Swing leg Formulation Model.

Where M is the inertia matrix, V is the Coriolis term, and G is the gravity term. The acceleration of inertia originating in the acceleration of the center of gravity of the whole robot is input to the acceleration of the root joint of the swing leg at the first step of the Newton-Euler method. We set it at $u_x, u_z - g$ in the Newton-Euler method for swing leg.

Equation(4.3) can be easily expanded to biped, quadruped, and another type of legged robot by appending nonlinear equations of motion of swing legs to the state equation of the center of gravity of the robot.

In this method,

(1) The nonlinearity of swing legs can be taken into account.
(2) The state equation is simple and practical.
(3) The equations for motion of swing legs are easily appended to state equations.

Therefore, this formulation can be extended to multi-legged robots. It enable us to make feasible formation.

4.2 Performance Index

Equation (4.4) is used as an evaluation function.

$$J = \phi[x(t+T)] + \tfrac{1}{2}\int_t^{t+T} \{(x^*(\tau)-x_f)^T \cdot Q \cdot (x^*(\tau)-x_f) + (u^*(\tau)-u_f)^T \cdot R \cdot (u^*(\tau)-u_f)\} d\tau \tag{4.4}$$

$$\phi[x^*(t+T)] = (x^*(t+T)-x_f)^T \cdot S_f \cdot (x^*(t+T)-x_f)$$

$$(u^*(\tau)-u_f)^T \cdot R \cdot (u^*(\tau)-u_f) = r_1 \cdot u_x^{*2}(\tau) + r_2 \cdot (u_z^*(\tau)-g)^2 + r_3 \cdot u_{zmp}^{*2}(\tau) + r_4 \cdot u_{\theta_1}^{*2}(\tau) + r_5 \cdot u_{\theta_2}^{*2}(\tau)$$

The first term is a terminal constraint. Including u_{zmp} in the performance index is a novel approach which allows us to obtain a solution that minimizes input of acceleration in each axis direction, input of each joint torque of swing leg, and norm of ZMP. It reduces sway of ZMP in the area of the robot sole (biped) or in the area of the polygon whose vertexes are described by the toes of the grounding legs (quadruped).

4.3 Slack Variable Method

The slack variable d(t) is introduced to make the control input explicit.

$$height + l_1 \cdot \sin\theta_1(t) + l_2 \cdot \sin(\theta_1(t) + \theta_2(t)) - d^2(t) = 0 \tag{4.5}$$

First, the inequality constraint is converted to an equal constraint equation. We described this as $S(x,t) = 0$, and differentiate it with respect to time until the control input appears in the equation.

$$\frac{dS}{dt} = \frac{\partial S}{\partial t} + \frac{\partial S}{\partial t}\dot{x} = \frac{\partial S}{\partial t} + \frac{\partial S}{\partial t} f(x,u) \tag{4.6}$$

In this case, the control input will appear by differentiating the equation twice.

$$l_1 \cdot \dot{\theta}_1(t) \cdot \cos\theta_1(t) + l_2 \cdot (\dot{\theta}_1(t) + \dot{\theta}_2(t)) \cdot \cos(\theta_1(t) + \theta_2(t)) - 2 \cdot d(t) \cdot \dot{d}(t) = 0 \tag{4.7}$$

$$\begin{aligned} & l_1 \cdot u_{\theta_1}(t) \cdot \cos\theta_1(t) + l_2 \cdot (u_{\theta_1}(t) + u_{\theta_2}(t)) \cdot \cos(\theta_1(t) + \theta_2(t)) \\ & -l_1 \cdot \dot{\theta}_1^2(t) \cdot \sin\theta_1(t) - l_2 \cdot (\dot{\theta}_1(t) + \dot{\theta}_2(t))^2 \cdot \sin(\theta_1(t) + \theta_2(t)) \\ & -2 \cdot \dot{d}^2(t) = 0 \end{aligned} \tag{4.8}$$

By introducing new state variables to the state equation,

$$\frac{d}{dt} d(t) = \dot{d}(t) \tag{4.9}$$

$$\frac{d}{dt} (\dot{d}(t)) = u_{slack}(t) \tag{4.10}$$

The equation can be written with the new input variable u_{slack} . The initial values d(t) and $\dot{d}(t)$ which satisfy equations **(**4.9**)** and **(**4.10**)** need to be determined. In the case of Table 3, d = 0.36, and $\dot{d}(t) = 0$

$$\frac{d}{dt} \begin{bmatrix} x(t) \\ z(t) \\ \theta_1(t) \\ \theta_2(t) \\ \dot{x}(t) \\ \dot{z}(t) \\ \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ d(t) \\ \dot{d}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \\ \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ u_x(t) \\ u_z(t) - g \\ u_{\theta_1}(t) \\ u_{\theta_2}(t) \\ \dot{d}(t) \\ u_{slack}(t) \end{bmatrix} \tag{4.11}$$

The number of state variables becomes ten on addition of two new variables, and the input variables become six by adding one. The ZMP condition can be described as follows.

$$u_{zmp}(t) \cdot (u_z(t) - g) = x(t) \cdot (u_z(t) - g) - z(t) \cdot u_x(t) \tag{4.12}$$

\subsection{Performance Index}
Equation(4.13) is used as an performance index.

$$\begin{aligned} J &= \phi[x(t+T)] + \tfrac{1}{2} \int_t^{t+T} \{(x(\tau) - x_f)^T \cdot Q \cdot (x(\tau) - x_f) + (u(\tau) - u_f)^T \cdot R \cdot (u(\tau) - u_f)\} d\tau \\ \phi[x(t+T)] &= (u(t+T) - u_f)^T \cdot S_f \cdot (u(t+T) - u_f) \end{aligned} \tag{4.13}$$

The first term is a terminal constraint. After the state variables are converged to the reference states, the inputs will be 0. Solving by RHC, the input of the Z-axis component will start to be generated again when the altitude of the center of gravity begins to drop. Major transition in the altitude causes major transition in its ZMP variable. To decrease these, we devised a method in which the terms related to u_z in the function of performance index are replaced by $u_z - g$ and a constant value is previously allocated to u_z. This method is effective for problems including the gravity term.

4.4 Numerical Calculation (Nonlinear Case)

We then formulated the problem, including the nonlinear term of the swing legs.

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ z(t) \\ \theta_1(t) \\ \theta_2(t) \\ \dot{x}(t) \\ \dot{z}(t) \\ \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ d(t) \\ \dot{d}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \\ \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ u_x(t) \\ u_z(t) - g \\ M^{-1}(\Theta)(u(t) - V(\Theta, \dot{\Theta}) - G(\Theta)) \\ \\ \dot{d}(t) \\ u_{slack}(t) \end{bmatrix} \tag{4.14}$$

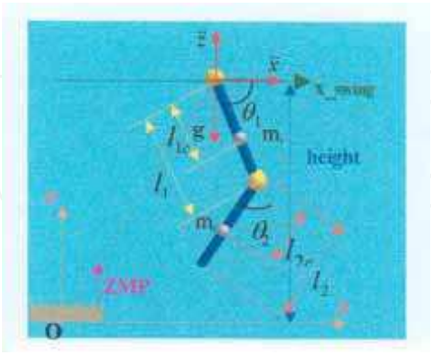
In this case, Equation(4.14) is substituted into $u_1(t)$ and $u_2(t)$ in Equation(4.15).

$$\begin{bmatrix} u_{\theta_1}(t) \\ u_{\theta_2}(t) \end{bmatrix} = M(\Theta) \cdot \ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta)) \tag{4.15}$$

The link parameters used in the calculation are listed in Fig. 13 and the parameters used in the optimization are shown in Table 3. The simulation time was set at 1.0s and its calculation took 0.9s. The simulation solution converged more rapidly than the case of linear. The control interval is set at 2.0ms. How much nonlinearity can be taken into account in the formulation depends on the capacity of the computer used. More nonlinearity can be included in the formulation if a higher performance computer is used.

S_f	diag[10.0,200.0,20.0,10.0,1.0$e^{-4},$,1.0e^{-4},1.0e^{-4},1.0e^{-4},1.0,1.0]
Q	diag[1.0,4.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0]
R	diag[1.0,1.0,1.0,1.0,10.0,1.0]
x_0	[0.0,1.0,-1.7,-0.6,0.27,0.0,0.36,0.0]^{T},
x_f	[0.2,1.0,-1.2,-1.0,0.27,0.0,0.0,0.0]^{T},
u_f	[0.0,9.8,0.0,0.0,0.1,0.0]^{T},
T_f	0.1s
Δt	2ms
Number of divide	5
ζ	450
height	1.0m

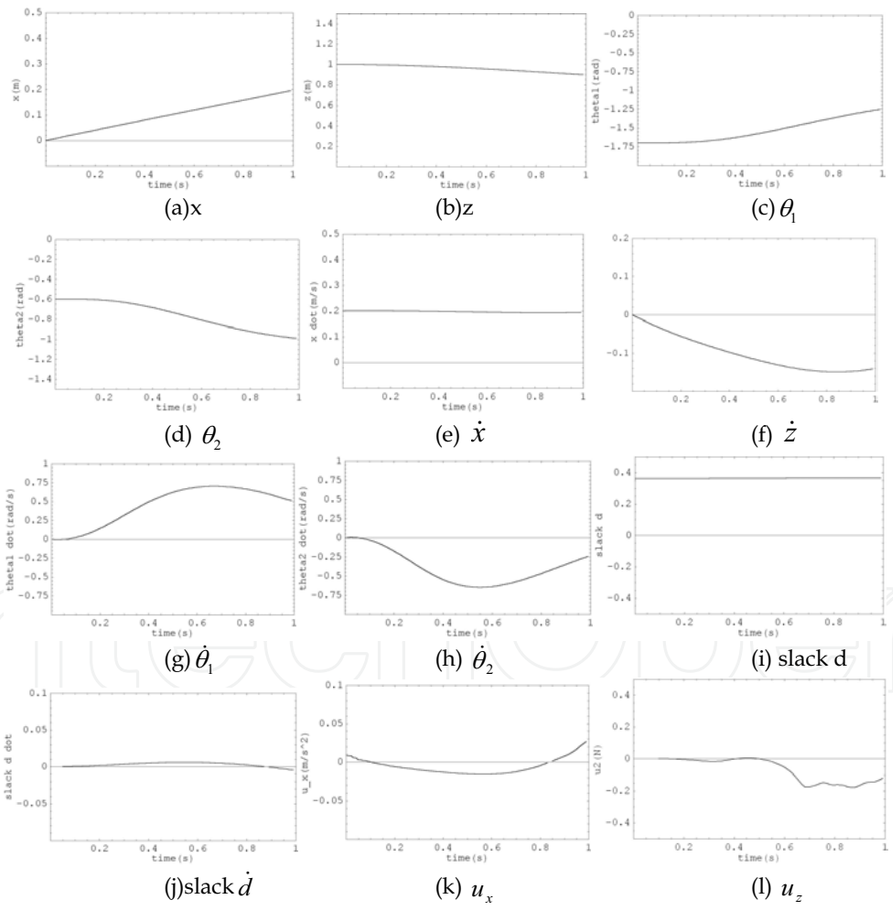
Table 3. Parameters for Simulation (Nonlinear Case).



link parameters

l_1	0.5m
l_2	0.5m
l_{1c}	0.5m
l_{2c}	0.5m
m_1	0.5kg
m_2	0.5kg

Fig. 13. The link model.

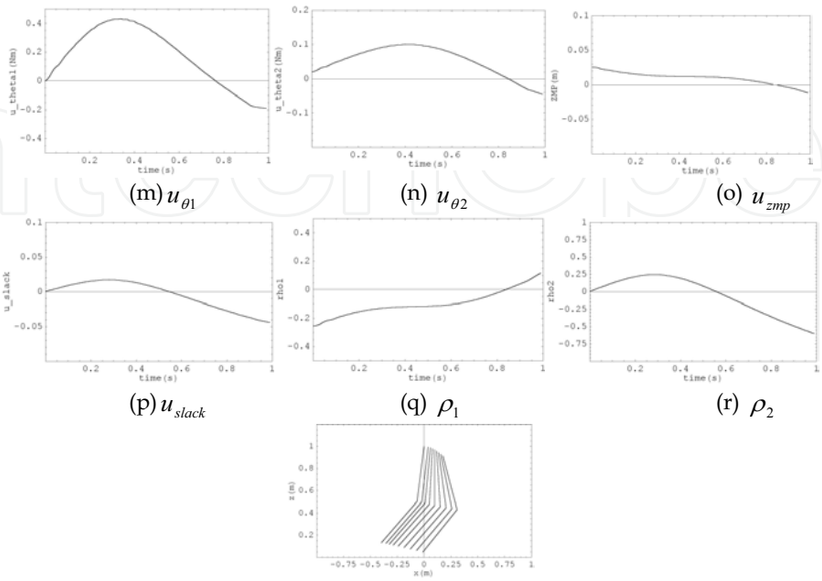


(a)x (b)z (c) θ_1

(d) θ_2 (e) $\dot{x}$ (f) $\dot{z}$

(g) $\dot{\theta}_1$ (h) $\dot{\theta}_2$ (i) slack d

(j)slack $\dot{d}$ (k) u_x (l) u_z



(m) $u_{\theta 1}$ (n) $u_{\theta 2}$ (o) u_{zmp}

(p) u_{slack} (q) ρ_1 (r) ρ_2

(s)Stick figure

Fig. 14. Simualtion(Slack Variable Method).

5. References

Bryson Jr, A. E., Ho, Y.C., Applied Optimal Control, Hemisphere, 1975

Hirai, K., Hirose, M., Haikawa, Y., Takenaka, T. (1998). The Development of Honda humanoid Robot, Proc. IEEE Int. Conf. Robotics and Automation, pp. 1321-1326

Mitobe, K., Masuyama, A., Shibata, T., Yamano, M., Nasu,Y. (2002). A ZMP Manipulation Method for Walking Robots and its Application to Angular Momentum Control, Journal of Robot Society of Japan Vol. 20, No. 5, pp. 53-58

Ohtsuka, T., Fujii, H. (1997). Real-Time Optimization Algorithm for Nonlinear Receding Horizon Control, Automatica, 33-6, p.p. 1147-1154

Vukobratovic, M., Juricic, D. (1969), Contributions to the synthesis of biped gait. IEEE Trans on Biomedical Engineering, BME-16:1-6

Vukobratovic, M., Frank, M. A. A., Juricic, D. (1970). On the Stability of Biped Locomotion, IEEE Trans on Biomedical Engineering, BME-17, No.1, pp. 25-36

D. Q. Mayne, H. Michalska, Receding Horizon Control of Nonlinear Systems, IEEE Trans, AC-35-7, pp. 814-824, 1990

Mayne, D. Q., Michalska, H., Robust receding horizon control of constrained nonlinear systems, IEEE Trans. on AC, Vol. 38, No. 11, pp. 1623-1633, 1993

J. L. Speyer andA. E. Bryson Optimal Programming Problems with a Bounded State Space, AIAA journal, vol. 6, p.p. 1488-1492,1968

Takeuchi, H. (2003). Real Time Optimization for Robot Control using Receding Horizon Control with Equal Constraint, Journal of Robotic Systems, Vol. 20, No. 1, p.p. 3-13

Takeuchi, H. (2004). Real Time Optimization and Control of Legged Robot - Formulization with Inequality State Constraint for - , Journal of Robotic Systems, Vol. 21, No. 4, p.p. 153-166

Mobile Robotics, Moving Intelligence

Edited by Jonas Buchli

ISBN 3-86611-284-X

Hard cover, 586 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers many aspects of the exciting research in mobile robotics. It deals with different aspects of the control problem, especially also under uncertainty and faults. Mechanical design issues are discussed along with new sensor and actuator concepts. Games like soccer are a good example which comprise many of the aforementioned challenges in a single comprehensive and in the same time entertaining framework. Thus, the book comprises contributions dealing with aspects of the Robotcup competition. The reader will get a feel how the problems cover virtually all engineering disciplines ranging from theoretical research to very application specific work. In addition interesting problems for physics and mathematics arises out of such research. We hope this book will be an inspiring source of knowledge and ideas, stimulating further research in this exciting field. The promises and possible benefits of such efforts are manifold, they range from new transportation systems, intelligent cars to flexible assistants in factories and construction sites, over service robot which assist and support us in daily live, all the way to the possibility for efficient help for impaired and advances in prosthetics.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen