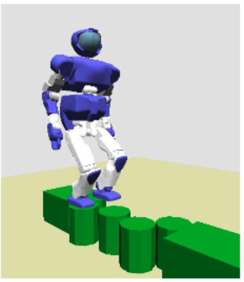
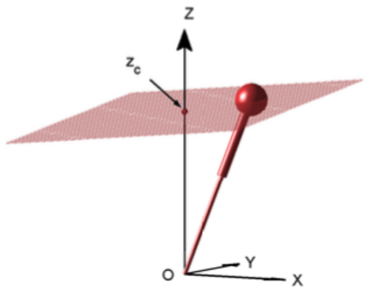
(MPC) 2.1 Paper Review: Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point

1. A biped walking pattern generation by using a preview control of the zero-moment point (ZMP).
2. The dynamics of a biped robot is modeled as a running cart on a table which gives a convenient representation to treat ZMP.
3. The paper formalizes the problem as the design of a ZMP tracking servo controller. Such controller adopts the preview control theory that uses the future reference.
4. It is also shown that a preview controller can be used to compensate the ZMP error caused by the difference between a simple model and the precise multibody model.

(MPC) 2.1.1 Dynamic Models of Biped Robot





1. Consider cart-table model to derive the dynamics on the plane $z = c_z$

$$\tau_y = -mgc_x + m\ddot{c}_x c_z$$
$$\tau_x = mgc_y - m\ddot{c}_y z_c$$

Rearranging them, we have

$$\ddot{c}_x = \frac{g}{c_z} c_x + \frac{1}{mc_z} \tau_y$$
$$\ddot{c}_y = \frac{g}{c_z} c_y - \frac{1}{mc_z} \tau_x$$

Let us introduce the ZMP

$$p_x = -\frac{\tau_y}{mg} \qquad \ddot{c}_x = \frac{g}{c_z}(c_x - p_x) \qquad p_x = c_x - \frac{c_z}{g}\ddot{c}_x$$
$$p_y = \frac{\tau_x}{mg} \qquad \ddot{c}_y = \frac{g}{c_z}(c_y - p_y) \qquad p_y = c_y - \frac{c_z}{g}\ddot{c}_y$$

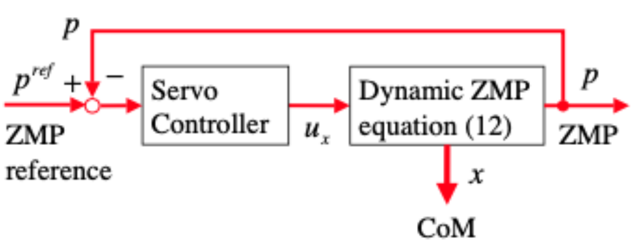
2. Choose the control term as derivative of the acceleration

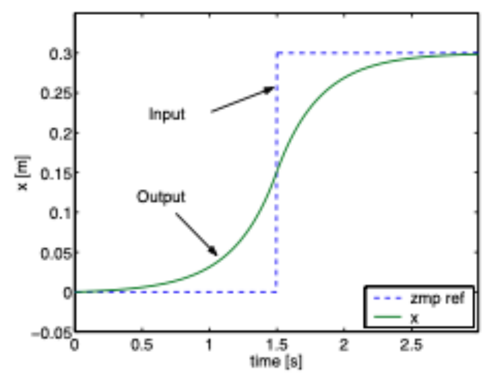
$$u_x = \frac{d\ddot{c}_x}{dt} \qquad\qquad u_y = \frac{d\ddot{c}_y}{dt}$$

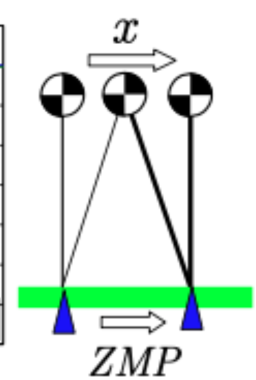
then we have the state space equation on x-directional ZMP equation

$$\frac{d}{dt}\begin{bmatrix} c_x \\ \dot{c}_x \\ \ddot{c}_x \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} c_x \\ \dot{c}_x \\ \ddot{c}_x \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x \qquad\qquad \dot{x}_m = A_c x_m + B_c u_x$$

$$p_x = \begin{bmatrix} 1 & 0 & -\frac{c_z}{g} \end{bmatrix}\begin{bmatrix} c_x \\ \dot{c}_x \\ \ddot{c}_x \end{bmatrix} + [0]u_x \qquad\qquad p_x = C_c x_m + D_c u_x$$







(MPC) 2.2 Paper Review: Design of an optimal controller for a discrete-time system subject to previewable demand

1. Let us convert the continuous system into discrete system using matlab commands

```
Delta_t = 0.005;
[Am,Bm,Cm,Dm] = c2dm(Ac,Bc,Cc,Dc,Delta_t)
```

2. Consider the time-invariant discrete system described by

$$x_m(k+1) = A_m x_m(k) + B_m u(k)$$
$$p(k) = C_m x_m(k)$$

3. We further assume that the demand is previewable in the sense that at each time k, N_l future values $p_r(k+1), p_r(k+2), \cdots, p_r(k+N_l)$ as well as present and past values of the demand are available.

4. It is desired to embed the integrator to eliminate the tracking error

$$e(k) = p(k) - p_r(k) = C_m x_m(k) - p_r(k)$$

   we derive an augmented state-space model that includes the future information on the demand signal as well as the error $e(i)$, the incremental state vector $\Delta x_m(k) = x_m(k) - x_m(k-1)$ and the incremental control vector $\Delta u(k) = u(k) - u(k-1)$

5. For the integrator embedding, taking a difference operation gives us

$$\begin{aligned} x_m(k+1) - x_m(k) &= A_m[x_m(k) - x_m(k-1)] + B_m[u(k) - u(k-1)] \\ e(k+1) - e(k) &= [C_m x_m(k+1) - p_d(k+1)] - [C_m x_m(k) - p_d(k)] \\ &= C_m(x_m(k+1) - x_m(k)) - p_r(k+1) + p_r(k) \end{aligned}$$

Let us denote the incremental demand defined by

$$\Delta p_r(k+1) = p_r(k+1) - p_r(k)$$

Now we have the below equation using the incremental notations:

$$\begin{aligned} \Delta x_m(k+1) &= A_m \Delta x_m(k) + B_m \Delta u(k) \\ e(k+1) &= e(k) + C_m \Delta x_m(k+1) - \Delta p_r(k+1) \\ &= e(k) + C_m[A_m \Delta x_m(k) + B_m \Delta u(k)] - \Delta p_r(k+1) \\ &= e(k) + C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) - \Delta p_r(k+1) \end{aligned}$$

6. Now we have an *augmented state-space model* as follow:

$$\begin{bmatrix} \Delta x_m(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) + \begin{bmatrix} 0_m \\ -1 \end{bmatrix} \Delta p_r(k+1)$$

$$\rightarrow \quad x(k+1) = Ax(k) + B\Delta u(k) + B_p \Delta p_r(k+1)$$

$$e(k) = \begin{bmatrix} 0_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ e(k) \end{bmatrix} \quad \rightarrow \quad e(k) = Cx(k)$$

7. The *future state variables* are calculated sequentially using the set of future control parame-

ters

$$
\begin{aligned}
x(k+1|k) &= Ax(k) + B\Delta u(k) + B_p\Delta p_r(k+1) \\
x(k+2|k) &= Ax(k+1|k) + B\Delta u(k+1) + B_p\Delta p_r(k+2) \\
&= A^2x(k) + AB\Delta u(k) + B\Delta u(k+1) + AB_p\Delta p_r(k+1) + B_p\Delta p_r(k+2) \\
x(k+3|k) &= Ax(k+2|k) + B\Delta u(k+2) + B_p\Delta p_r(k+3) \\
&= A^3x(k) + A^2B\Delta u(k) + AB\Delta u(k+1) + B\Delta u(k+2) \\
&\quad + A^2B_p\Delta p_r(k+1) + AB_p\Delta p_r(k+2) + B_p\Delta p_r(k+3) \\
&\vdots \\
x(k+N_p|k) &= A^{N_p}x(k) + A^{N_p-1}B\Delta u(k) + A^{N_p-2}B\Delta u(k+1) + \cdots + A^{N_p-N_c}B\Delta u(k+N_c-1) \\
&\quad + A^{N_p-1}B_p\Delta p_r(k+1) + A^{N_p-2}B_p\Delta p_r(k+2) + \cdots + A^{N_p-N_l}B_p\Delta p_r(k+N_l)
\end{aligned}
$$

8. From the predicted state variables, the *predicted error variables* are, by substitution

$$
\begin{aligned}
e(k+1|k) &= CAx(k) + CB\Delta u(k) + CB_p\Delta p_r(k+1) \\
e(k+2|k) &= CAx(k+1|k) + CB\Delta u(k+1) + CB_p\Delta p_r(k+2) \\
&= CA^2x(k) + CAB\Delta u(k) + CB\Delta u(k+1) + CAB_p\Delta p_r(k+1) + CB_p\Delta p_r(k+2) \\
e(k+3|k) &= CAx(k+2|k) + CB\Delta u(k+2) + CB_p\Delta p_r(k+3) \\
&= CA^3x(k) + CA^2B\Delta u(k) + CAB\Delta u(k+1) + CB\Delta u(k+2) \\
&\quad + CA^2B_p\Delta p_r(k+1) + CAB_p\Delta p_r(k+2) + CB_p\Delta p_r(k+3) \\
&\vdots \\
e(k+N_p|k) &= CA^{N_p}x(k) + CA^{N_p-1}B\Delta u(k) + CA^{N_p-2}B\Delta u(k+1) + \cdots + CA^{N_p-N_c}B\Delta u(k+N_c-1) \\
&\quad + CA^{N_p-1}B_p\Delta p_r(k+1) + CA^{N_p-2}B_p\Delta p_r(k+2) + \cdots + CA^{N_p-N_l}B_p\Delta p_r(k+N_l)
\end{aligned}
$$

9. Note that all predicted error variables are formulated in terms of current state $x(k)$, the future control movement $\Delta u(k+j)$ for $j = 0, 1, 2, \cdots, N_c - 1$, and the demand $\Delta p_r(k+j)$ for $j = 1, 2, 3, \cdots, N_l$. As a compact form,

$$E = Fx(k) + \Phi\Delta U + \Psi\Delta P_r$$

$$\begin{bmatrix} e(k+1|k) \\ e(k+2|k) \\ e(k+3|k) \\ \vdots \\ e(k+N_p|k) \end{bmatrix} = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix} x(k) + \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-2}B & \cdots & CA^{N_p-N_c}B \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}$$

$$+ \begin{bmatrix} CB_p & 0 & 0 & \cdots & 0 \\ CAB_p & CB & 0 & \cdots & 0 \\ CA^2B_p & CAB_p & CB_p & \cdots & 0 \\ \vdots \\ CA^{N_p-1}B_p & CA^{N_p-2}B_p & CA^{N_p-2}B_p & \cdots & CA^{N_p-N_l}B_p \end{bmatrix} \begin{bmatrix} \Delta p_r(k+1) \\ \Delta p_r(k+2) \\ \Delta p_r(k+3) \\ \vdots \\ \Delta p_r(k+N_l) \end{bmatrix}$$

10. This compact form will be utilized for the implementation of the MPC.

$$E = Fx(k) + \Phi\Delta U + \Psi\Delta P_r$$

where $E \in \Re^{N_p}$, $F \in \Re^{N_p \times n}$, $x(k) \in \Re^n$, $\Phi \in \Re^{N_p \times N_c}$, $\Delta U \in \Re^{N_c}$, $\Psi \in \Re^{N_p \times N_l}$, $\Delta P_r \in \Re^{N_l}$

11. This objective is then translated into a design to find the *best* control parameter vector ΔU such that an error function. Let us define the cost function J that reflects the control objective

$$J = \frac{1}{2} E^T Q E + \frac{1}{2} \Delta U^T \bar{R} \Delta U$$

where the error weighting $Q \in \Re^{N_p \times N_p}$ is defined according to the prediction sequence level and the control input weighting $\bar{R} = r_w I_{N_c \times N_c}$ is a diagonal matrix and r_w is a *tuning* parameter.

$$Q = \begin{bmatrix} \frac{1}{\log(1+1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{\log(2+1)} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \frac{1}{\log(N_p+1)} \end{bmatrix} \qquad \bar{R} = \begin{bmatrix} r_w & 0 & \cdots & 0 \\ 0 & r_w & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & r_w \end{bmatrix}$$

12. To find the optimal ΔU that will minimize J,

$$\begin{aligned} J &= \frac{1}{2}(Fx(k) + \Phi \Delta U + \Psi \Delta P_r)^T Q (Fx(k) + \Phi \Delta U + \Psi \Delta P_r) + \frac{1}{2} \Delta U^T \bar{R} \Delta U \\ &= \frac{1}{2}(Fx(k) + \Psi \Delta P_r)^T Q (Fx(k) + \Psi \Delta P_r) + \Delta U^T \Phi^T Q (Fx(k) + \Psi \Delta P_r) + \frac{1}{2} \Delta U^T \Phi^T Q \Phi \Delta U + \frac{1}{2} \Delta U^T \bar{R} \Delta U \end{aligned}$$

13. The *necessary condition* of the minimum J is obtained as

$$\frac{\partial J}{\partial \Delta U} = \Phi^T Q (Fx(k) + \Psi \Delta P_r) + \Phi^T Q \Phi \Delta U + \bar{R} \Delta U = 0 \quad \rightarrow \quad \Delta U = -(\Phi^T Q \Phi + \bar{R})^{-1} \Phi^T Q (Fx(k) + \Psi \Delta P_r)$$

where the matrix $(\Phi^T Q \Phi + \bar{R})$ is called the *Hessian* matrix in the optimization literature.

14. The optimal solution of the control signal is linked to the demand $p_d(k+i)$ and the state variable $x(k)$:

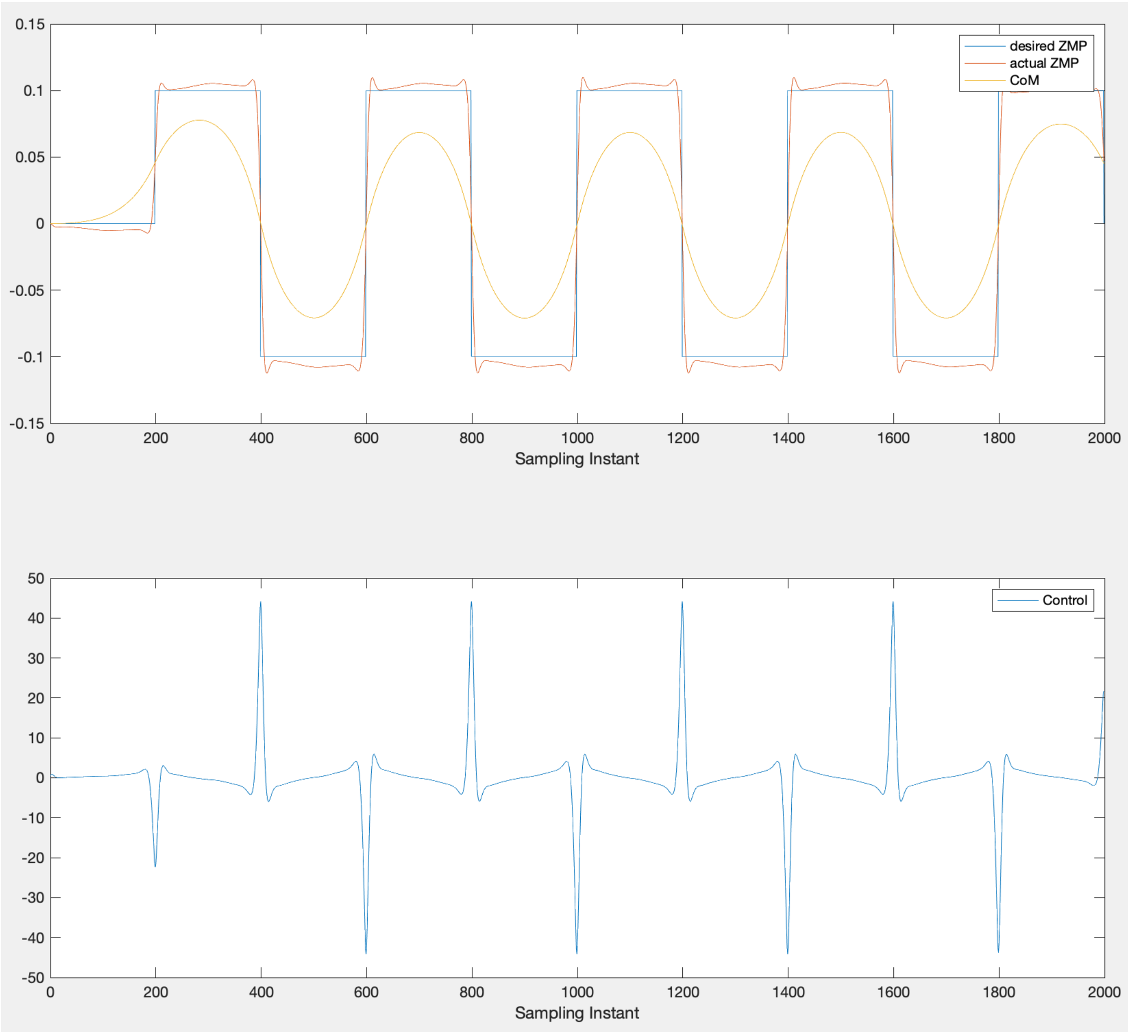
$$\begin{aligned}\Delta U &= -(\Phi^T Q \Phi + \bar{R})^{-1} \Phi^T Q (F x(k) + \Psi \Delta P_r) \\ &= -K_{mpc} x(k) - K_{preview} \Delta P_r\end{aligned}$$

where

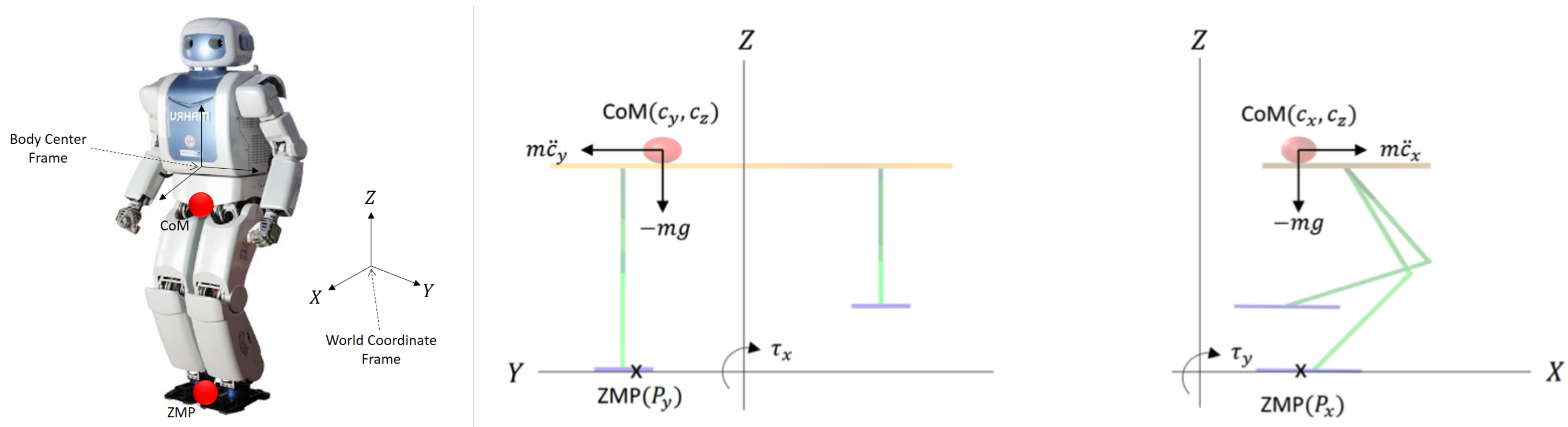
$$\begin{aligned}K_{mpc} &= (\Phi^T Q \Phi + \bar{R})^{-1} \Phi^T Q F \\ K_{preview} &= (\Phi^T Q \Phi + \bar{R})^{-1} \Phi^T Q \Psi\end{aligned}$$

15. According to the receding horizon principle, the first sample $\Delta u(k)$ is chosen from a optimal solution ΔU

$$u(k) = u(k-1) + \Delta u(k)$$


desired ZMP
actual ZMP
CoM
Sampling Instant
Control
Sampling Instant

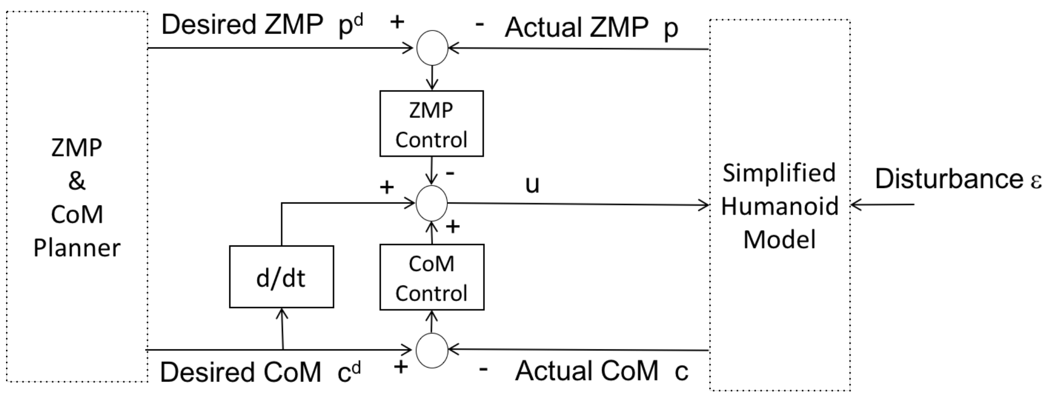
(MPC) 2.3 Paper Review: Posture/Walking Control of Humanoid Robot based on the Kinematic Resolution of CoM Jacobian with Embedded Motion



1. Consider two differential equations related to the ZMP as follows:

$$p_x = c_x - \frac{c_z}{g}\ddot{c}_x = c_x - \frac{1}{\omega_n^2}\ddot{c}_x \qquad p_y = c_y - \frac{c_z}{g}\ddot{c}_y = c_y - \frac{1}{\omega_n^2}\ddot{c}_y$$

where $\omega_n = \sqrt{g/c_z}$ implies a *natural radian frequency* of the simplified motion model of humanoid.



2. The simultaneous CoM and ZMP stabilizing control scheme is suggested for humanoid balancing as shown in the figure. The ZMP and CoM Planner yields the desired ZMP and CoM trajectories using the MPC based on previewable demand.

$$p_i^d = c_i^d - \frac{1}{\omega_n^2}\ddot{c}_i^d \qquad \text{for } \; i = x, y \tag{155}$$

where the superscript d implies the desired trajectory.

3. The simplified motion model for humanoid robot has the following dynamics:

$$p_i = c_i - \frac{1}{\omega_n^2}\ddot{c}_i \qquad\qquad \dot{c}_i = u_i + \varepsilon_i \tag{156}$$

where ε_i is the bounded disturbance, u_i is a control input to be designed later for the CoM motion control.

4. Let us define the *ZMP and CoM errors* from the figure as follows:

$$e_{pi} = p_i^d - p_i \qquad\qquad e_{ci} = c_i^d - c_i \tag{157}$$

5. Using the concept of proportional control in the ZMP and CoM control blocks, we are to design the control input u_i in the figure as following form:

$$u_i = \dot{c}_i^d + k_{ci} e_{ci} - k_{pi} e_{pi} \qquad \text{for } i = x, y. \tag{158}$$

where k_{ci} and k_{pi} are the proportional gains of CoM control and ZMP control, respectively.

6. Stability of the proposed control scheme.

   a) The error dynamics is obtained from the difference between Eqs. (155) and (156) as follows:

   $$\ddot{e}_{ci} = \omega_n^2 (e_{ci} - e_{pi}). \tag{159}$$

   b) Another error dynamics is obtained by applying Eq. (158) to Eq. (156) as follows:

   $$\dot{e}_{ci} + k_{ci} e_{ci} + \varepsilon_i = k_{pi} e_{pi}. \tag{160}$$

   c) Differentiating Eq. (160) and using Eq. (159) yield the following dynamic relation:

   $$\begin{aligned} \dot{e}_{pi} &= \frac{1}{k_{pi}} \left(\ddot{e}_{ci} + k_{ci} \dot{e}_{ci} + \dot{\varepsilon}_i \right) \\ &= \frac{\omega_n^2}{k_{pi}} (e_{ci} - e_{pi}) + \frac{k_{ci}}{k_{pi}} (k_{pi} e_{pi} - k_{ci} e_{ci} - \varepsilon_i) + \frac{1}{k_{pi}} \dot{\varepsilon}_i \\ &= \left(\frac{\omega_n^2 - k_{ci}^2}{k_{pi}} \right) e_{ci} - \left(\frac{\omega_n^2 - k_{pi} k_{ci}}{k_{pi}} \right) e_{pi} + \frac{1}{k_{pi}} (\dot{\varepsilon}_i - k_{ci} \varepsilon_i). \end{aligned} \tag{161}$$

   d) Let us take a Lyapunov function candidate as follow:

   $$V(e_{ci}, e_{pi}) = \frac{1}{2} \left[(k_{ci}^2 - \omega_n^2) e_{ci}^2 + k_{pi}^2 e_{pi}^2 \right], \tag{162}$$

where $V(e_{ci}, e_{pi})$ must be a positive definite function if $k_{ci} > \omega_n$ and $k_{pi} > 0$, except the equilibrium point $e_{ci} = 0$ and $e_{pi} = 0$.

e) If we take time differentiation with respect to above Lyapunov function candidate, then we have

$$\dot{V} = (k_{ci}^2 - \omega_n^2)e_{ci}\dot{e}_{ci} + k_{pi}^2 e_{pi}\dot{e}_{pi}.$$

By applying Eqs. (160) and (161) to above relation, we have

$$\begin{aligned}
\dot{V} =& - k_{ci}(k_{ci}^2 - \omega_n^2)e_{ci}^2 - k_{pi}(\omega_n^2 - k_{pi}k_{ci})e_{pi}^2 - (k_{ci}^2 - \omega_n^2)e_{ci}\varepsilon_i + k_{pi}e_{pi}\dot{\varepsilon}_i - k_{pi}k_{ci}e_{pi}\varepsilon_i \\
=& - k_{ci}(k_{ci}^2 - \omega_n^2)e_{ci}^2 - k_{pi}(\omega_n^2 - k_{pi}k_{ci})e_{pi}^2 \\
&+ (k_{ci}^2 - \omega_n^2)\left(\alpha^2 e_{ci}^2 - \left|\alpha e_{ci} + \frac{1}{2\alpha}\varepsilon_i\right|^2 + \frac{1}{4\alpha^2}\varepsilon_i^2\right) \\
&+ k_{pi}\left(\beta^2 e_{pi}^2 - \left|\beta e_{pi} - \frac{1}{2\beta}\dot{\varepsilon}_i\right|^2 + \frac{1}{4\beta^2}\dot{\varepsilon}_i^2\right) \\
&+ k_{pi}k_{ci}\left(\gamma^2 e_{pi}^2 - \left|\gamma e_{pi} + \frac{1}{2\gamma}\varepsilon_i\right|^2 + \frac{1}{4\gamma^2}\varepsilon_i^2\right) \\
=& - (k_{ci} - \alpha^2)(k_{ci}^2 - \omega_n^2)e_{ci}^2 - k_{pi}[\omega_n^2 - (k_{pi} + \gamma^2)k_{ci} - \beta^2]e_{pi}^2 \\
&- (k_{ci}^2 - \omega_n^2)\left|\alpha e_{ci} + \frac{1}{2\alpha}\varepsilon_i\right|^2 - k_{pi}\left|\beta e_{pi} - \frac{1}{2\beta}\dot{\varepsilon}_i\right|^2 - k_{pi}k_{ci}\left|\gamma e_{pi} + \frac{1}{2\gamma}\varepsilon_i\right|^2 \\
&+ \left[\frac{(k_{ci}^2 - \omega_n^2)}{4\alpha^2} + \frac{k_{pi}k_{ci}}{4\gamma^2}\right]\varepsilon_i^2 + \frac{k_{pi}}{4\beta^2}\dot{\varepsilon}_i^2,
\end{aligned}$$

where α, β, γ are arbitrary real positive constants. Now we can have the following inequality:

$$\dot{V} \leq -(k_{ci} - \alpha^2)(k_{ci}^2 - \omega_n^2)e_{ci}^2 - k_{pi}[\omega_n^2 - (k_{pi} + \gamma^2)k_{ci} - \beta^2]e_{pi}^2 + \left[\frac{(k_{ci}^2 - \omega_n^2)}{4\alpha^2} + \frac{k_{pi}k_{ci}}{4\gamma^2}\right]\varepsilon_i^2 + \frac{k_{p,i}}{4\beta^2}\dot{\varepsilon}_i^2. \tag{163}$$

f) If the following conditions are satisfied with $\alpha = \sqrt{\omega_n}$,

$$k_{ci} > \omega_n \quad \text{and} \quad 0 < k_{pi} < \left(\frac{\omega_n^2 - \beta^2}{k_{ci}} - \gamma^2\right) \tag{164}$$

then we can know that the *disturbance input-to-state stability* (ISS) is guaranteed

g) For practical use, the gain conditions of Eq. (164) can be approximately rewritten without using arbitrary positive constants β and γ as follows:

$$k_{ci} > \omega_n \quad \text{and} \quad 0 < k_{pi} < \omega_n \tag{165}$$

h) Notice that the positive feedback of actual ZMP information is used, while the CoM control is a typical negative feedback.

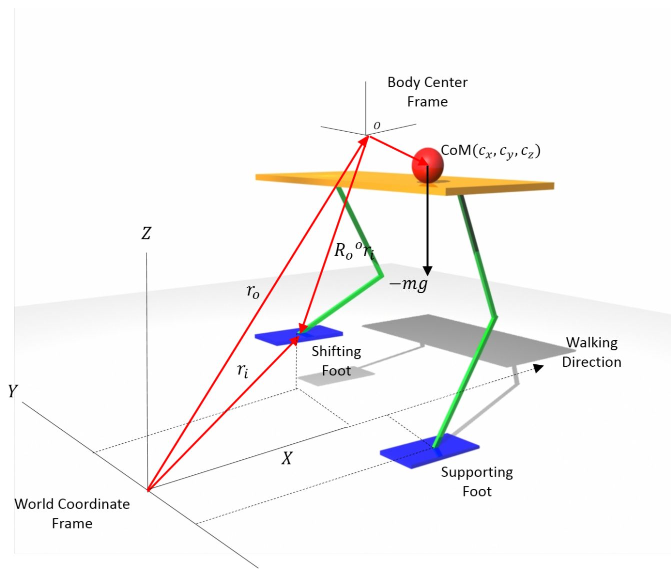
7. Since the control input produced from the simultaneous CoM and ZMP stabilizer is designed at the velocity level of CoM motion in the task space, the *kinematic resolution method* from the control input in the task space to the configuration joint velocities is required for the implementation.

8. Each limb of humanoid is, hereafter, considered as an independent limb. In general, ith limb has the following kinematic relation:

$$^o\dot{x}_i = {}^oJ_i\,\dot{q}_i \quad \text{for} \;\; i = 1, 2, \cdots, n \tag{166}$$

where $^o\dot{x}_i \in \Re^6$ is the velocity of end point of ith limb, $\dot{q}_i \in \Re^{n_i}$ is the joint velocity of ith limb, $^oJ_i \in \Re^{6\times n_i}$ is an usual Jacobian matrix of ith limb.

9. The leading superscript o implies that the elements are expressed in the body center coordinate frame. Note that the body center frame is attached and fixed on the humanoid robot.



10. Compatibility Condition

- Since the body center frame is floating due to the robotic movement, we have

$$r_i = r_o + R_o{}^or_i, \tag{167}$$

for $i = 1, 2, 3, 4$, where R_o is a rotation matrix of the body center frame with respect to the world coordinate frame.

- A 6-dimensional velocity vector of ith limb motion represented on the world coordinate frame is obtained as:

$$\begin{aligned} \dot{r}_i &= \dot{r}_o - [R_o{}^o r_i \times]\omega_o + R_o{}^o\dot{r}_i \\ \omega_i &= \omega_o + R_o{}^o\omega_i, \end{aligned}$$

as a matrix-vector form, we have

$$\begin{bmatrix} \dot{r}_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} I_3 & -[R_o{}^o r_i \times] \\ 0_3 & I_3 \end{bmatrix} \begin{bmatrix} \dot{r}_o \\ \omega_o \end{bmatrix} + \begin{bmatrix} R_o & 0_3 \\ 0_3 & R_o \end{bmatrix} \begin{bmatrix} {}^o\dot{r}_i \\ {}^o\omega_i \end{bmatrix}$$

where 0_3 and I_3 imply (3×3) zero and identity matrices, respectively.

- Above equation can be rewritten as following compact form:

$$\begin{aligned} \dot{x}_i &= X_i^{-1}\dot{x}_o + X_o{}^o\dot{x}_i, \\ &= X_i^{-1}\dot{x}_o + X_o{}^o J_i\ \dot{q}_i, \end{aligned} \tag{168}$$

where $\dot{x}_i = [\dot{r}_i^T, {\omega_i}^T]^T \in \Re^6$ implies a 6-dimensional velocity vector of end point of ith limb in terms of world coordinate frame, $\dot{x}_o = [\dot{r}_o^T, {\omega_o}^T]^T$ implies the velocity of body center frame, ${}^o\dot{x}_i = [{}^o\dot{r}_i^T, {}^o{\omega_i}^T]^T$ implies the velocity of end point of ith limb in terms of the body center frame, and we know ${}^o\dot{x}_i = {}^oJ_i\ \dot{q}_i$, and

$$X_i = \begin{bmatrix} I_3 & [R_o{}^o r_i \times] \\ 0_3 & I_3 \end{bmatrix} \in \Re^{6\times 6} \quad \text{and} \quad X_o = \begin{bmatrix} R_o & 0_3 \\ 0_3 & R_o \end{bmatrix} \in \Re^{6\times 6}$$

Hereafter we will use the relation $J_i = X_o{}^oJ_i$ for notational ease.

- Now we are ready to explain the compatibility condition. It means that all the limbs in humanoid should have the same body center velocity, in other words, from Eq. (168), we can see that all the limbs should satisfy the compatibility condition that the body center velocity is the same, and thus, ith and jth limbs should satisfy the following relation:

$$X_i(\dot{x}_i - J_i\dot{q}_i) = X_j(\dot{x}_j - J_j\dot{q}_j). \tag{169}$$

- Let us express the base limb with the subscript 1, then the joint velocity of ith limb is expressed as:

$$\dot{q}_i = J_i^+\dot{x}_i - J_i^+X_{i1}(\dot{x}_1 - J_1\dot{q}_1), \tag{170}$$

  for $i = 2, 3, 4$, where J_i^+ means the Moore-Penrose pseudoinverse of J_i and

$$X_{i1} = X_i^{-1}X_1 = \begin{bmatrix} I_3 & [R_o({}^or_1 - {}^or_i)\times] \\ 0_3 & I_3 \end{bmatrix}.$$

11. The derivation procedures for CoM Position c and CoM Jacobian J_{ci} of ith limb will be skipped. You can refer to the paper for the details.

12. Kinematic Resolution Method using CoM Jacobian

- The motion of body center frame can be obtained by using Eq. (168) for the base limb as follows:

$$\begin{aligned} \dot{x}_o &= X_1\left\{\dot{x}_1 - J_1\dot{q}_1\right\} \\ \begin{bmatrix} \dot{r}_o \\ \omega_o \end{bmatrix} &= \begin{bmatrix} I_3 & [R_o{}^or_1\times] \\ 0_3 & I_3 \end{bmatrix} \left\{ \begin{bmatrix} \dot{r}_1 \\ \omega_1 \end{bmatrix} - \begin{bmatrix} J_{v_1} \\ J_{\omega_1} \end{bmatrix} \dot{q}_1 \right\}, \end{aligned} \tag{171}$$

where J_{v_1} and J_{ω_1} are linear and angular velocity part of the base limb Jacobian J_1 expressed on the world coordinate frame, respectively.

- The CoM motion (or velocity) is described as following form:

$$\begin{aligned}\dot{c} &= \dot{r}_o + \omega_o \times (c - r_o) + \sum_{i=1}^{n} J_{ci}\ \dot{q}_i \\ &= \dot{r}_o + \omega_o \times (c - r_o) + J_{c1}\dot{q}_1 + \sum_{i=2}^{n} J_{ci}\ \dot{q}_i \\ &= \dot{r}_o + \omega_o \times (c - r_o) + J_{c1}\dot{q}_1 + \sum_{i=2}^{n} J_{ci}[J_i^+\dot{x}_i - J_i^+ X_{i1}\dot{x}_1 + J_i^+ X_{i1} J_1 \dot{q}_1)] \\ &= \dot{r}_o + \omega_o \times (c - r_o) + J_{c1}\dot{q}_1 + \sum_{i=2}^{n} J_{ci}J_i^+(\dot{x}_i - X_{i1}\dot{x}_1) + \sum_{i=2}^{n} J_{ci}J_i^+ X_{i1} J_1 \dot{q}_1. \end{aligned} \tag{172}$$

- Using above equations, the CoM motion is only related with the motion of base limb:

$$\dot{c} = \dot{r}_1 + \omega_1 \times r_{c1} - J_{v_1}\dot{q}_1 + r_{c1} \times J_{\omega_1}\dot{q}_1 + J_{c1}\dot{q}_1 + \sum_{i=2}^{n} J_{ci}J_i^+(\dot{x}_i - X_{i1}\dot{x}_1) + \sum_{i=2}^{n} J_{ci}J_i^+ X_{i1} J_1 \dot{q}_1 \tag{173}$$

where $r_{c1} = c - r_1$. In addition, let us assume that the base limb has the surface contact with the ground (the end point of base limb represented on the world coordinate frame is fixed, $\dot{x}_1 = 0$, namely, $\dot{r}_1 = 0$, $\omega_1 = 0$).

- Let us rearrange above equation by gathering the terms related to the base limb motion

$(\dot{q}_1)$, then it is reduced to:

$$\dot{c} - \sum_{i=2}^{n} J_{ci} J_i^+ \dot{x}_i = \left[-J_{v_1} + r_{c1} \times J_{\omega_1} + J_{c1} + \sum_{i=2}^{n} J_{ci} J_i^+ X_{i1} J_1 \right] \dot{q}_1, \tag{174}$$

where we can know from the left-hand side terms that all task motions of other limbs except base limb seem to be embedded into the CoM motion.

- Finally, the CoM Jacobian matrix having the embedded task motions can be defined like usual kinematic Jacobian with respect to the base limb:

$$\dot{c}_{fsem} = J_{fsem}\, \dot{q}_1, \tag{175}$$

where the subscript 'fsem' denotes a fully specified embedded task motion and

$$\dot{c}_{fsem} = \dot{c} - \sum_{i=2}^{n} J_{ci} J_i^+ \dot{x}_i, \tag{176}$$

$$J_{fsem} = -J_{v_1} + r_{c1} \times J_{\omega_1} + J_{c1} + \sum_{i=2}^{n} J_{ci} J_i^+ X_{i1} J_1. \tag{177}$$

- Here, if the CoM Jacobian having the embedded task motions is augmented with the orientation Jacobian of body center $(\omega_o = -J_{\omega_1}\dot{q}_1)$, then the desired joint configurations of base limb (support limb) are kinematically resolved as follows:

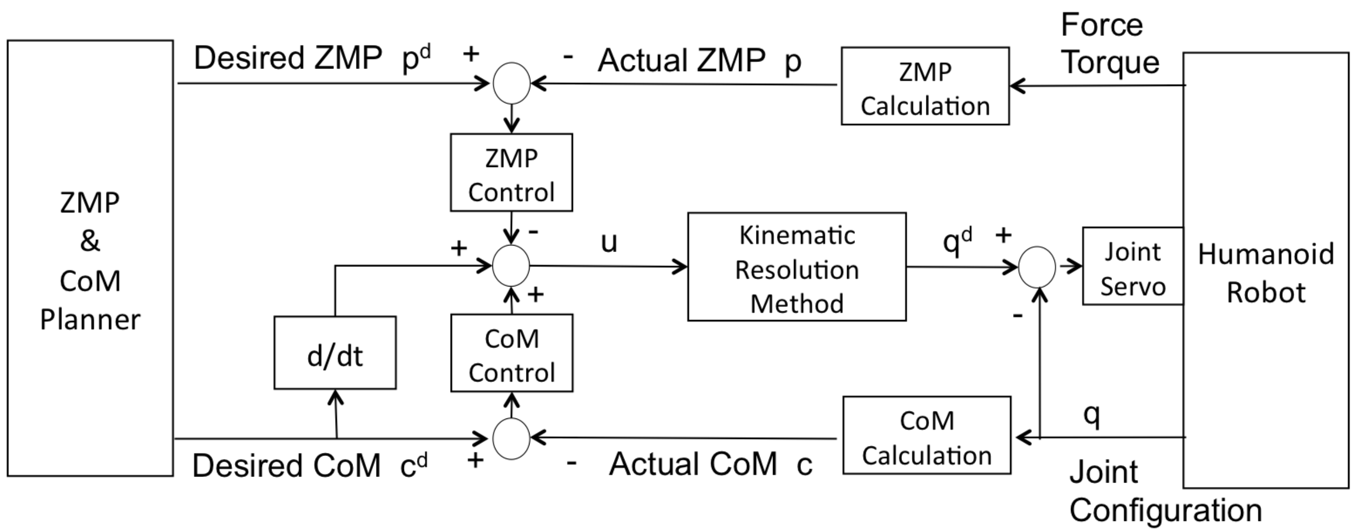
$$\dot{q}_{1,d} = \begin{bmatrix} J_{fsem} \\ -J_{\omega_1} \end{bmatrix}^+ \begin{bmatrix} \dot{c}_{fsem,d} \\ \omega_{o,d} \end{bmatrix}, \tag{178}$$

where the subscript d means the desired motion and

$$\dot{c}_{fsem,d} = \dot{c}_d - \sum_{i=2}^{n} J_{ci} J_i^+ \dot{x}_{i,d}. \tag{179}$$

- With the desired joint motion of base limb, $\dot{q}_{1,d}$, the desired joint motions of all other limbs can be obtained from the compatibility condition as follow:

$$\dot{q}_{i,d} = J_i^+(\dot{x}_{i,d} + X_{i1} J_1 \dot{q}_{1,d}), \qquad \text{for } i = 2, \cdots, n. \tag{180}$$



13. Related Experimental Videos

Balancing, https://www.youtube.com/watch?v=t-G9AoYD0vA/
Pushing-recovery, https://www.youtube.com/watch?v=HMsMXR_1cdY/
Pushing-recovery, https://www.youtube.com/watch?v=zO1uptT4g2I/
Pushing-recovery, https://www.youtube.com/watch?v=8-iJfTICmT8/
Dancing, https://www.youtube.com/watch?v=BTfXk_uipdc/
Walking, https://www.youtube.com/watch?v=v2ckoiD6qXY/