Modeling and Control of Legged Robots
SS 2023

L7: Walking

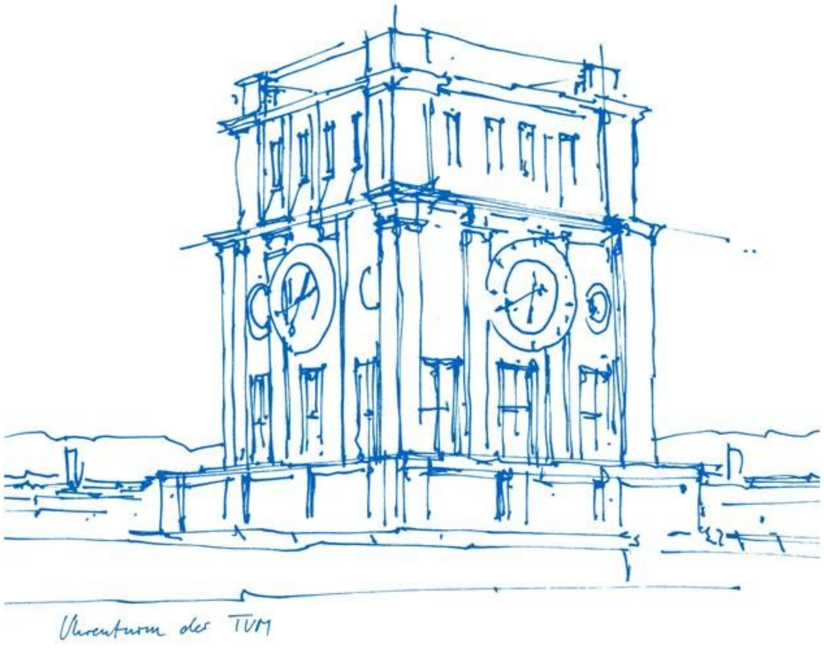
Dr.-Ing. J. Rogelio Guadarrama Olvera

Technical University of Munich

School of Computation, Information and Technology

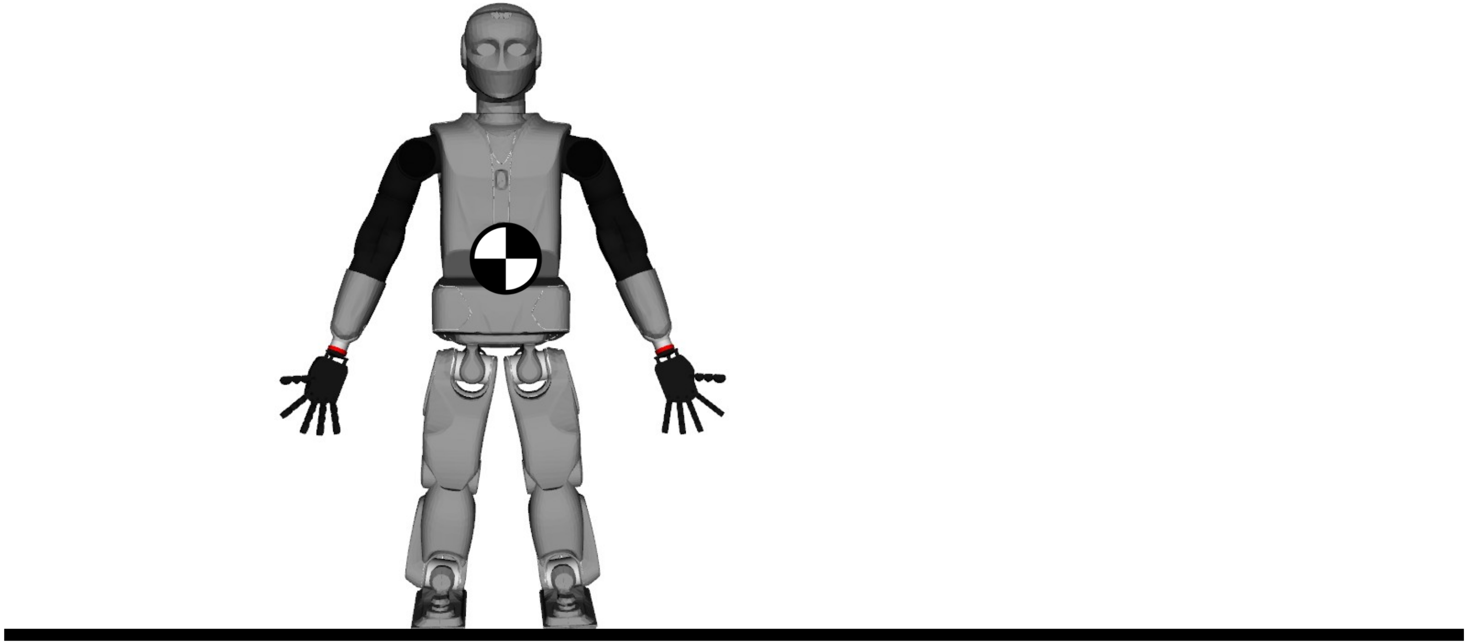
Chair of Cognitive Systems

Munich 18. June 2024

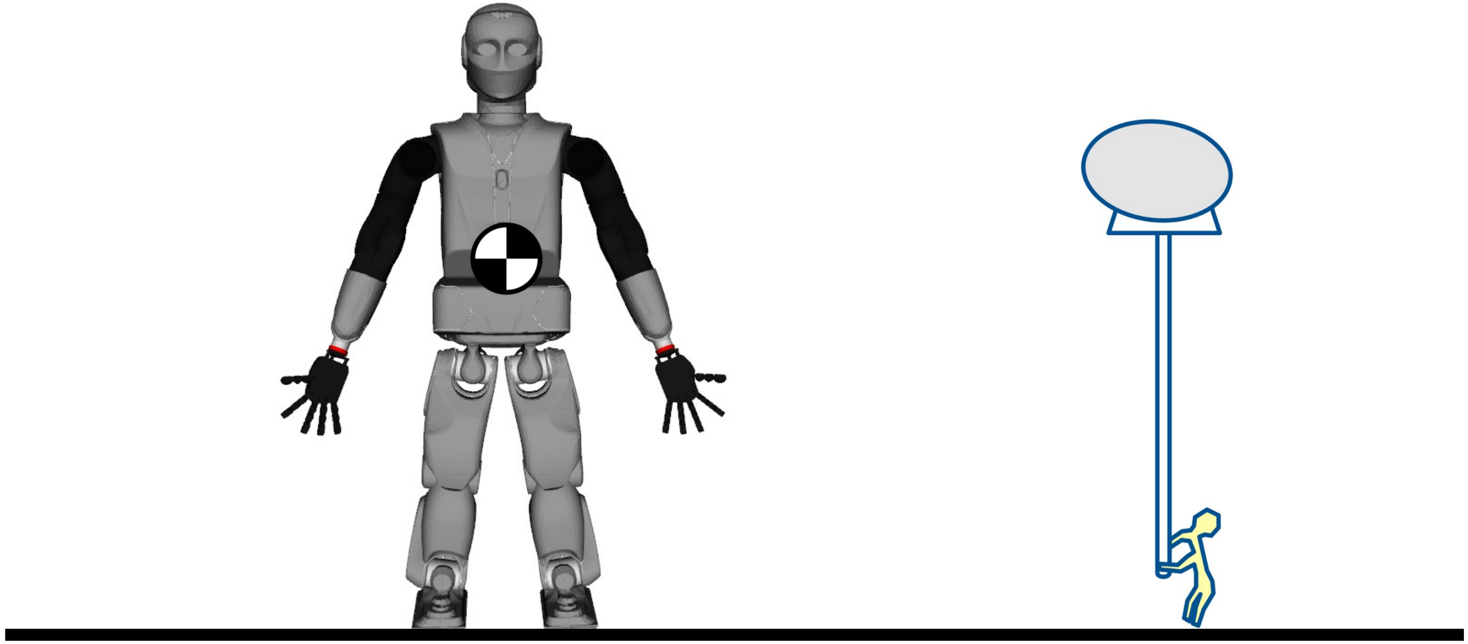


Contents

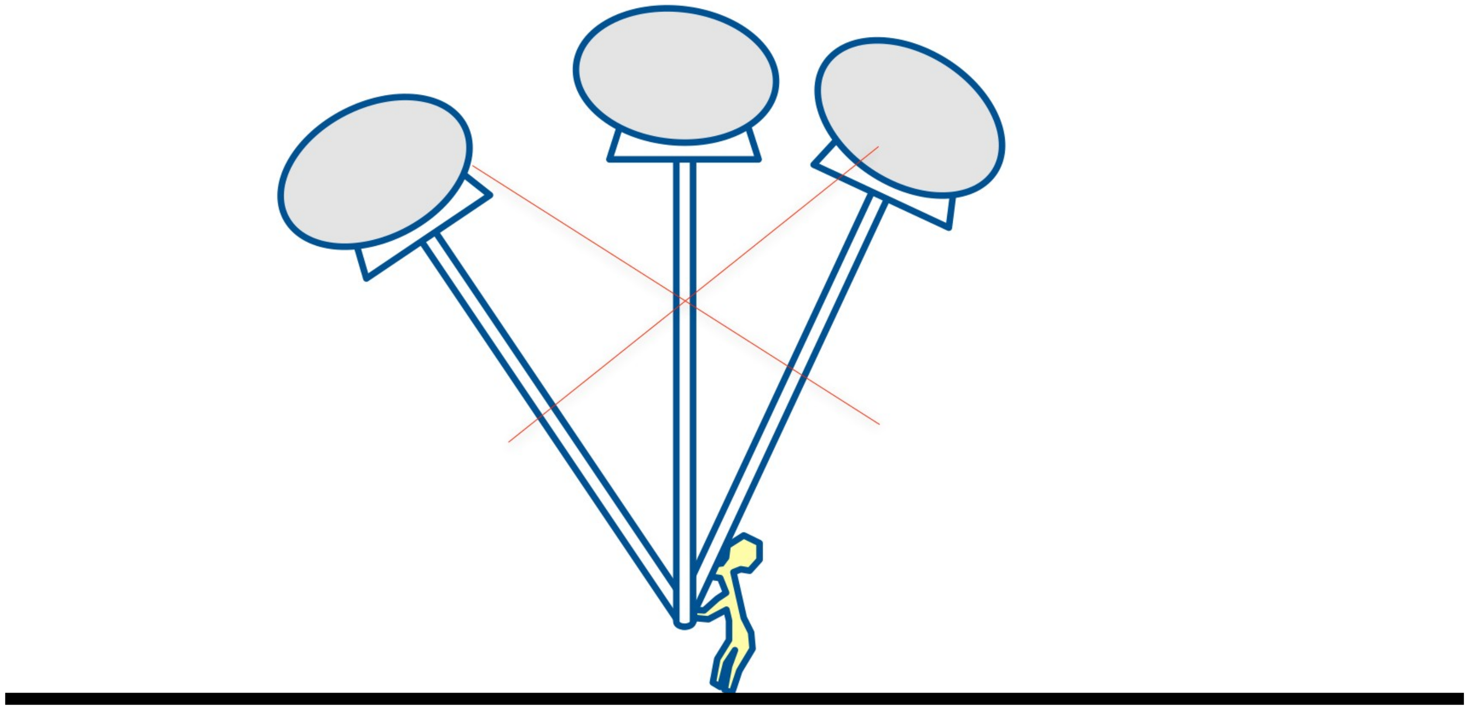
Introduction



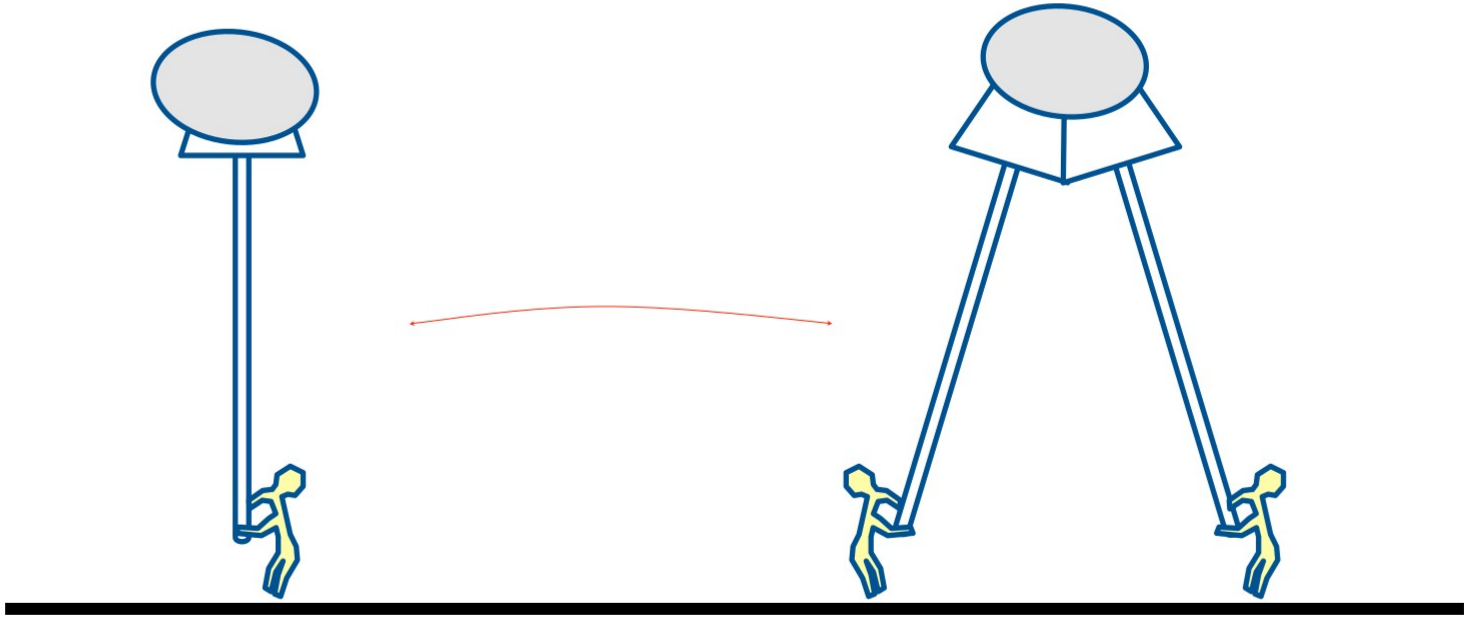
Introduction



Introduction

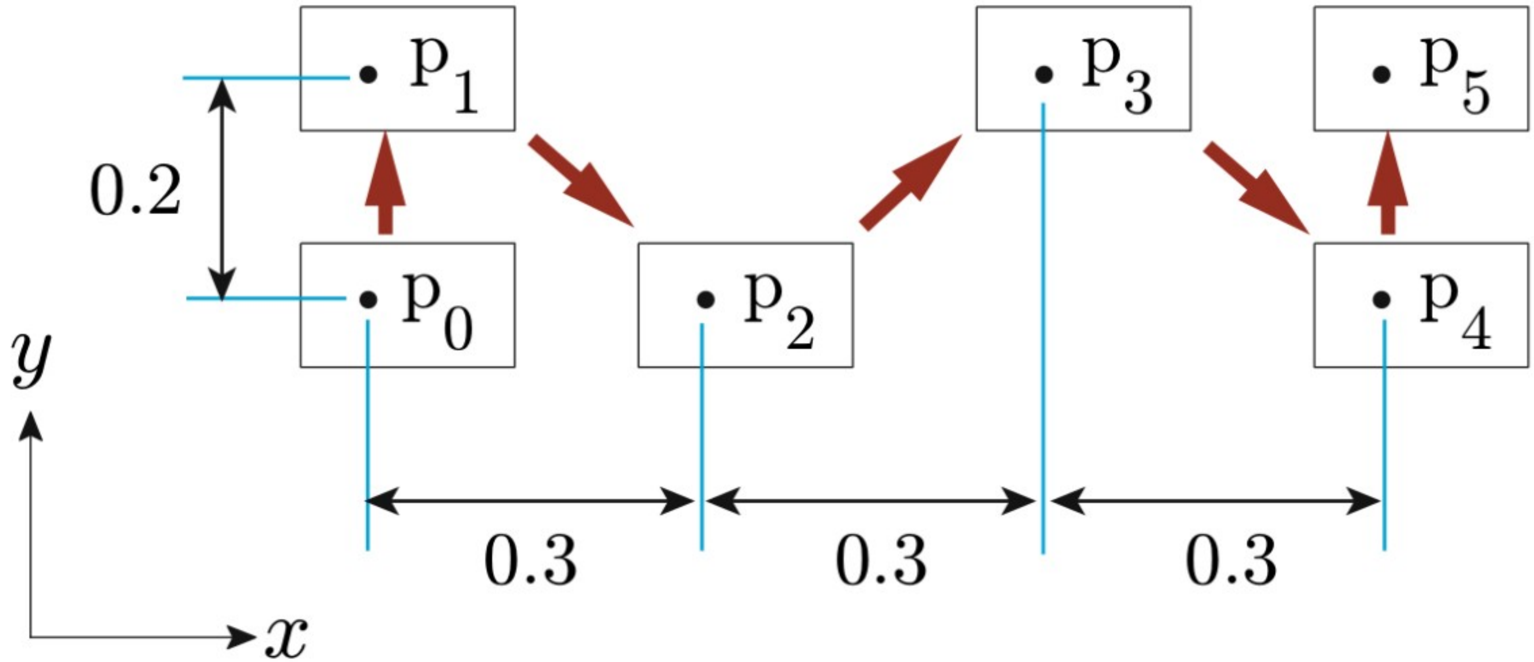


Introduction



Introduction

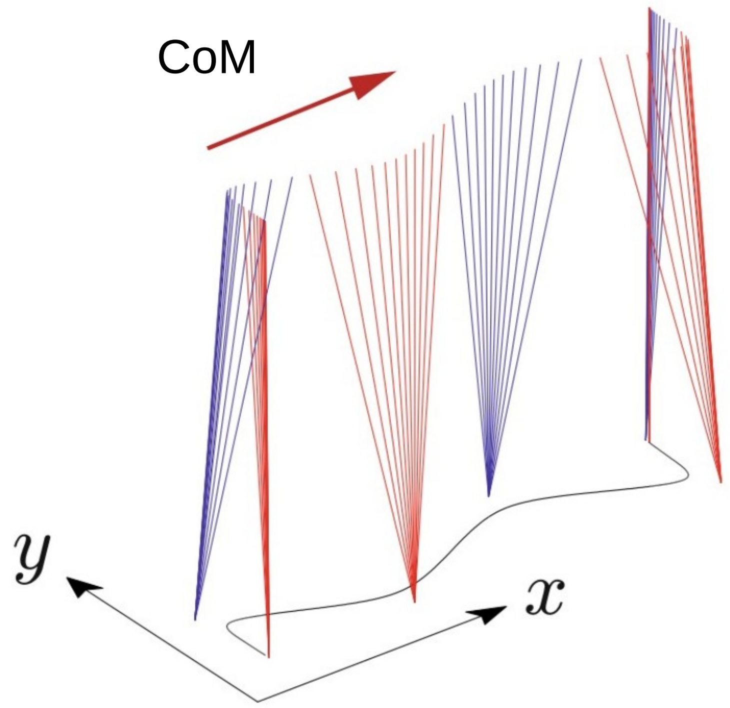
Walking as a Linear Inverted Pendulum (LIPM)



Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Introduction

Walking as a Linear Inverted Pendulum (LIPM)



Walking

Oxford Advanced Learner’s Dictionary, Oxford University Press:
Move along at a moderate pace by lifting up and putting down each foot in turn, so that one foot is on the ground while the other is being lifted

Walking

Oxford Advanced Learner's Dictionary, Oxford University Press:
Move along at a moderate pace by lifting up and putting down each foot in turn, so that one foot is on the ground while the other is being lifted

Walking

Walking

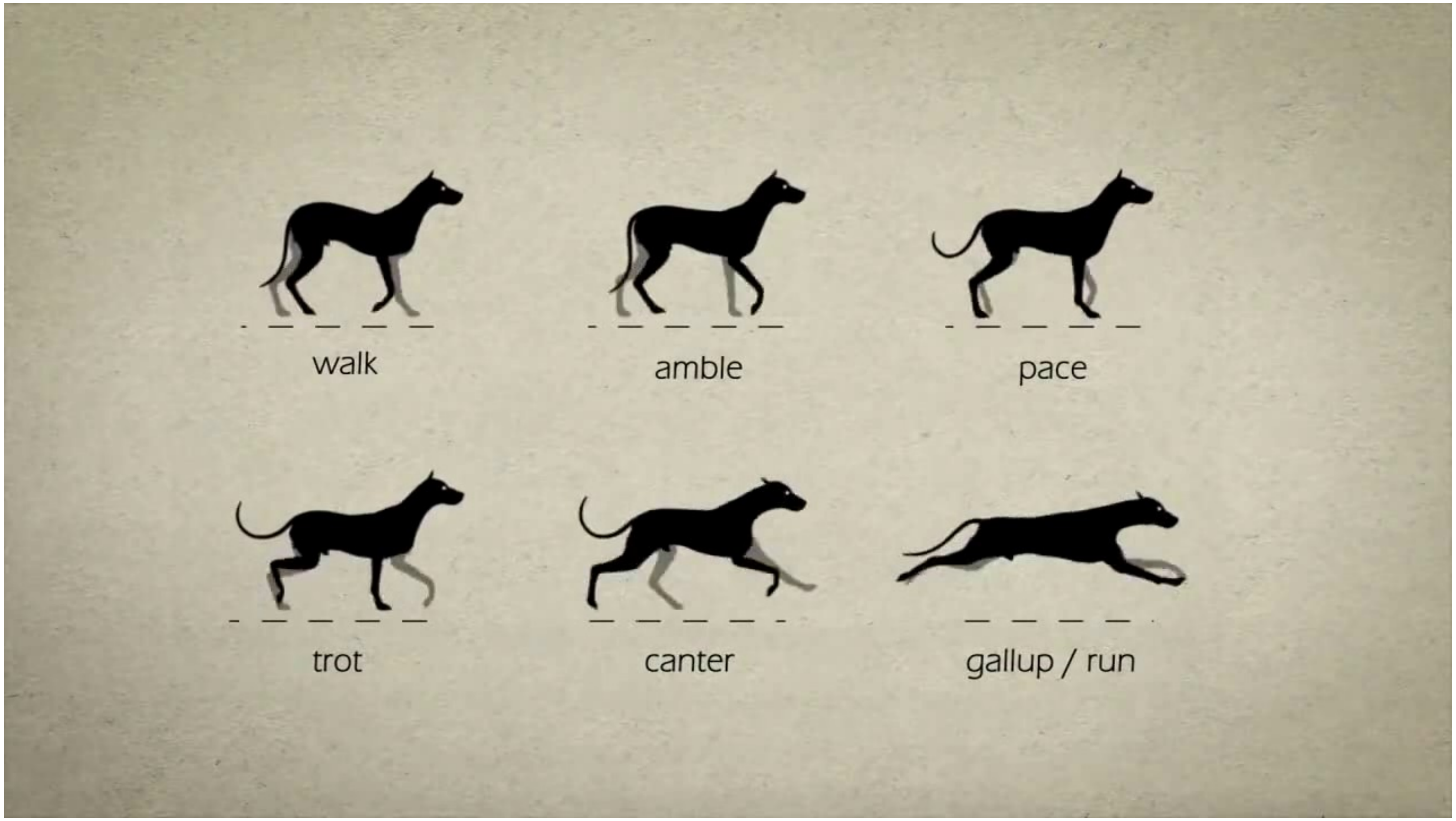
Oxford Advanced Learner's Dictionary, Oxford University Press:
Move along at a moderate pace by lifting up and putting down each foot in turn, so that one foot is on the ground while the other is being lifted

Walking

Running

Walking

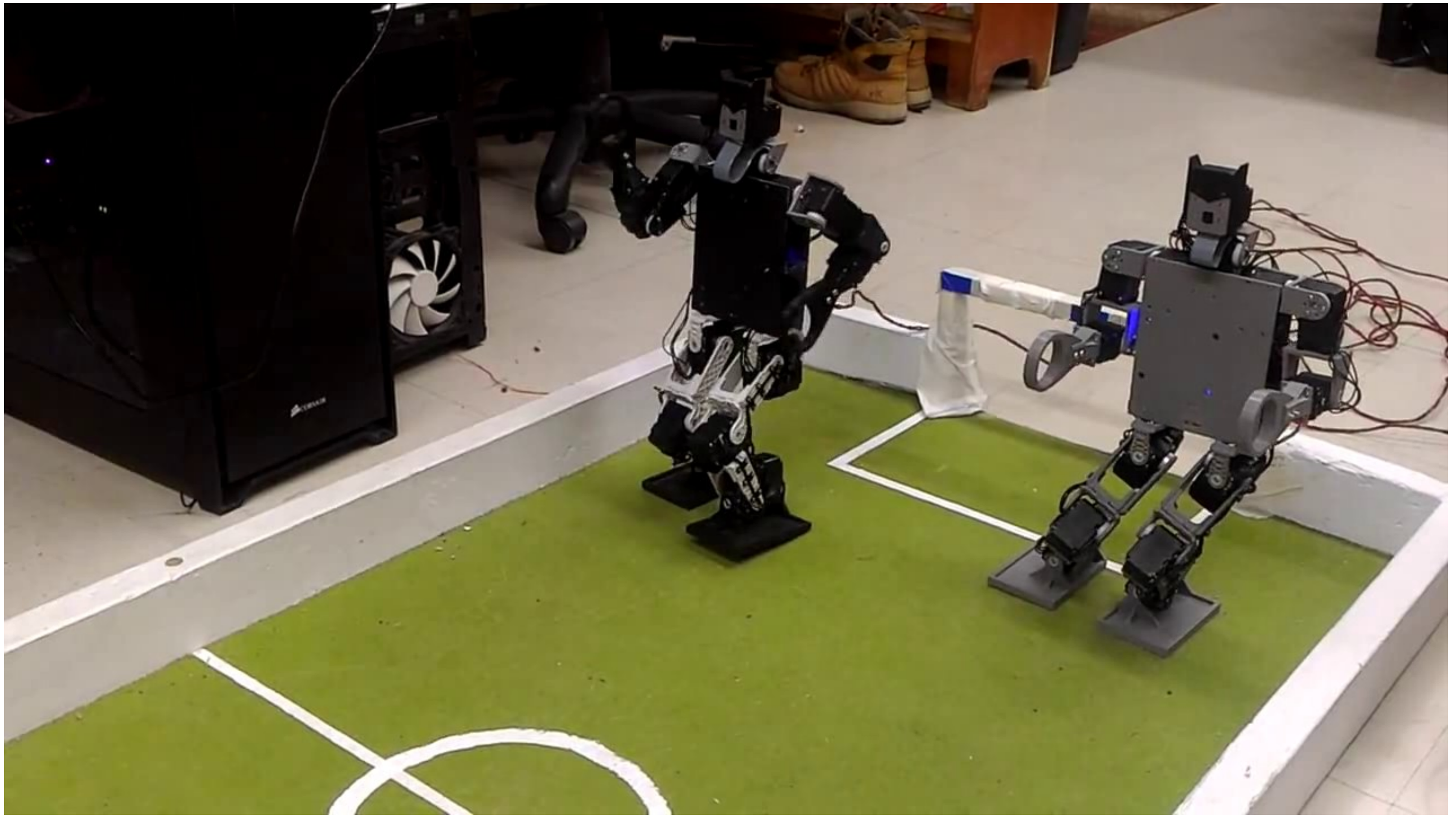
Legged locomotion can take different modalities. For example, Quadrupeds can:



Jamie Capsolas, Animal Gaits for Animators
https://www.youtube.com/watch?v=AZGNjKolAiQ

Walking

For bipeds



Dynamic walking

Static Walking

Walking

For bipeds

Dynamic Walking

- The projection of the CoM on the ground can leave the supporting polygon.
- Posture is dynamically stable.
- Will fall if you freeze the motors.
- High energy efficiency

Static Walking

- The projection of the CoM on the ground never leaves the supporting polygon.
- Posture is always stable.
- Will not fall if you freeze the motors.
- Not efficient.

Walking

For bipeds

- The projection of the CoM on the ground can leave the supporting polygon.
- Posture is dynamically stable.
- Will fall if you freeze the motors.
- High energy efficiency

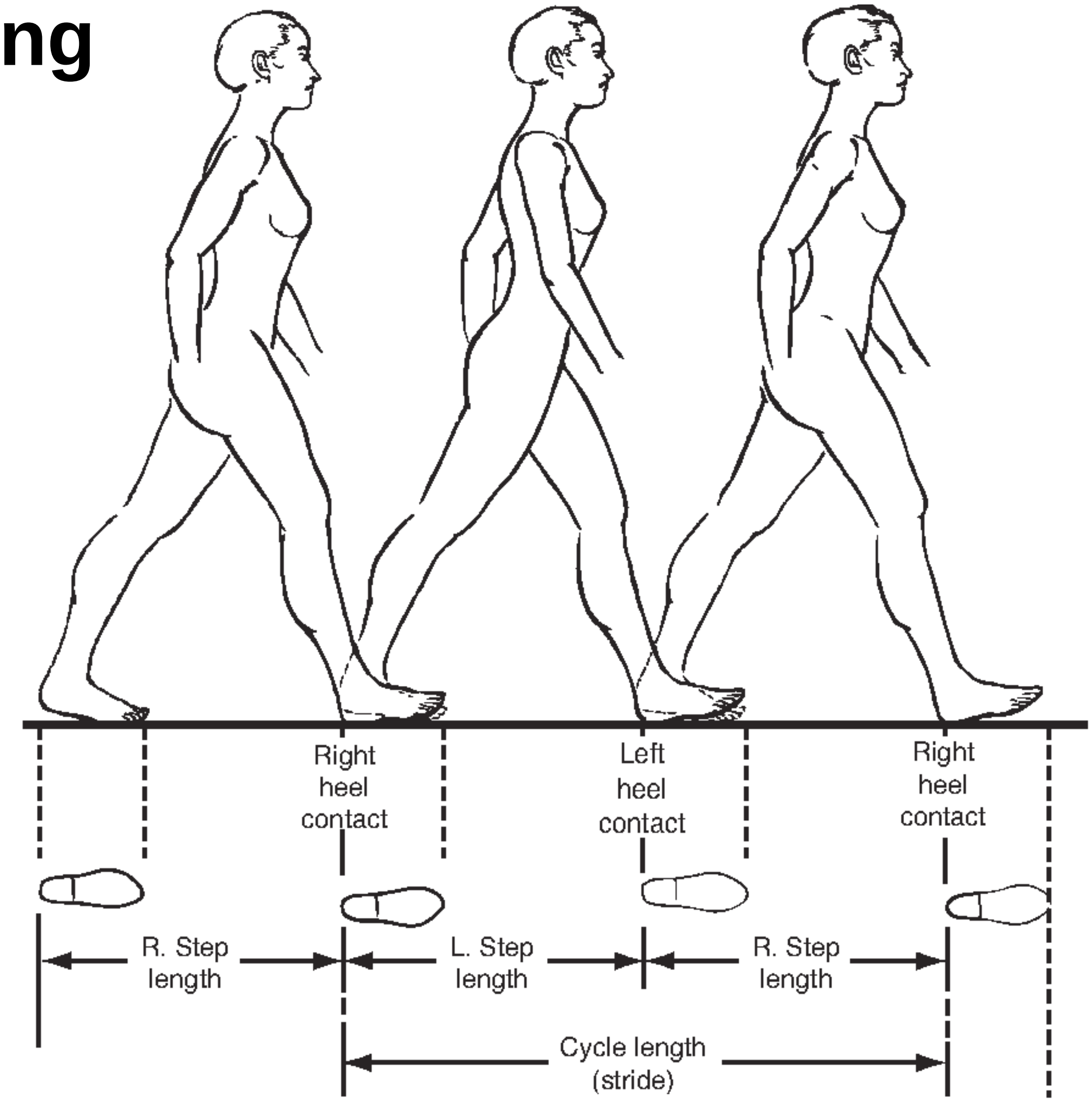
Dynamic Walking

- The projection of the CoM on the ground never leaves the supporting polygon.
- Posture is always stable.
- Will not fall if you freeze the motors.
- Not efficient.

Static Walking

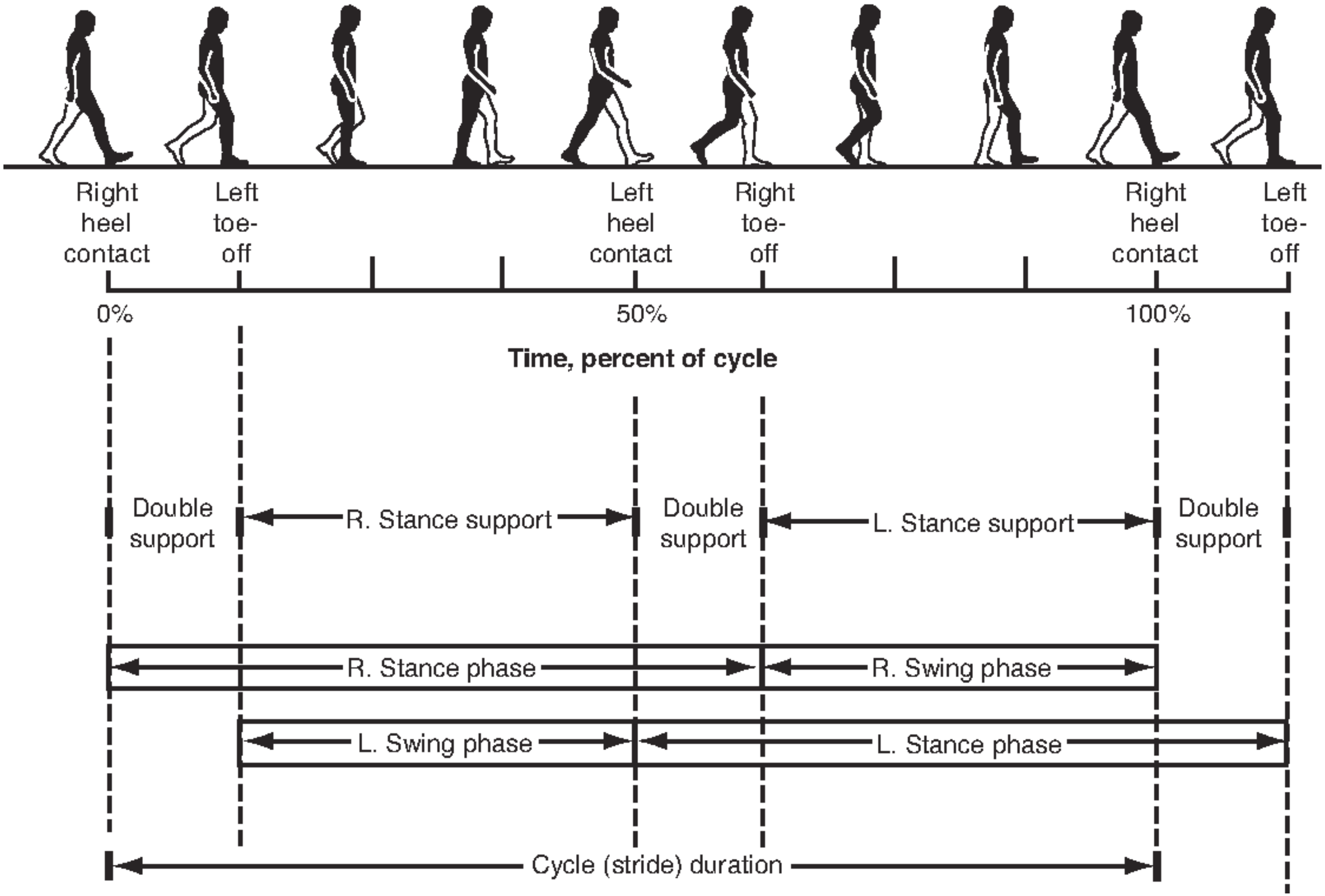
Human Walking Cycle

Space domain

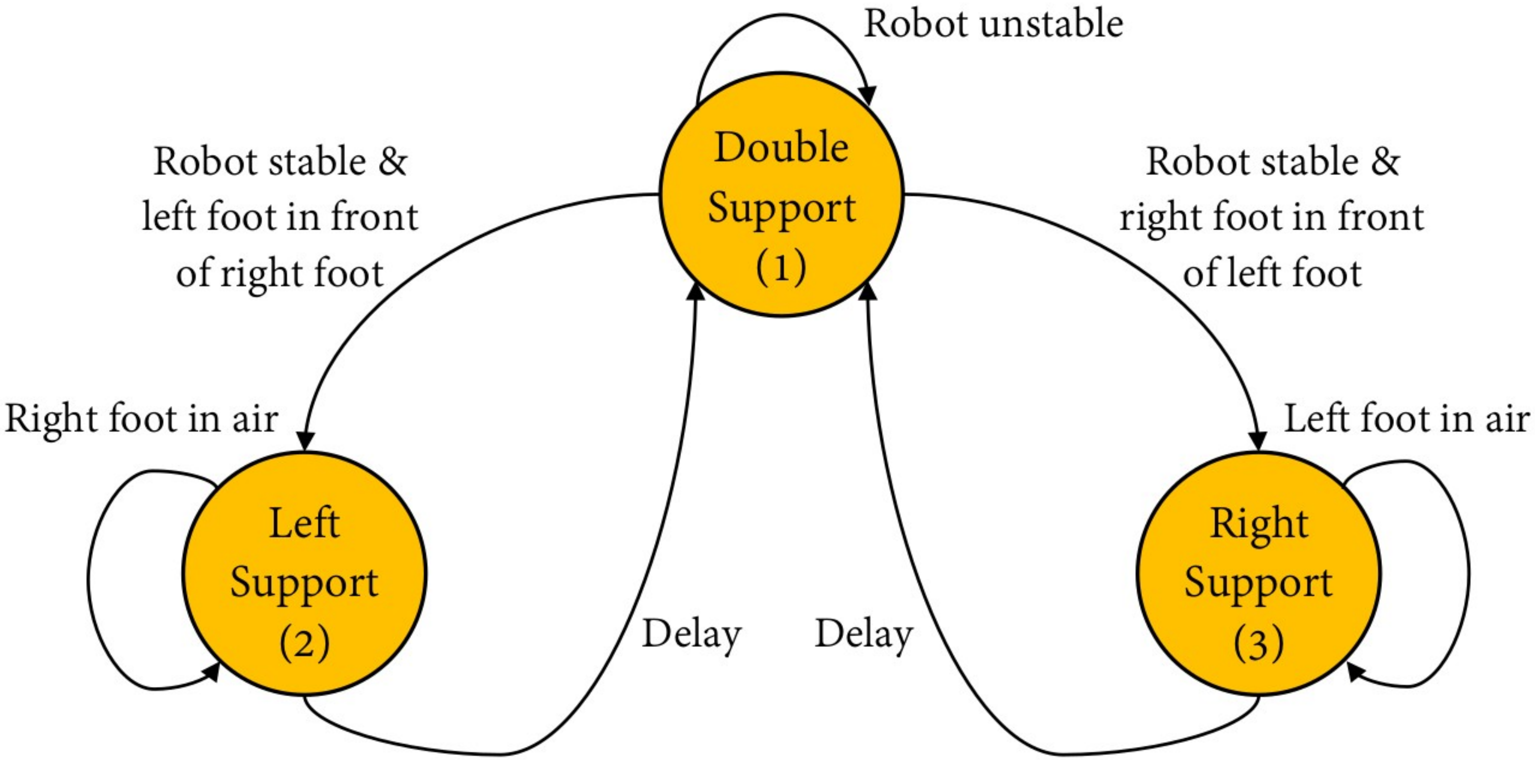


Human Walking Cycle

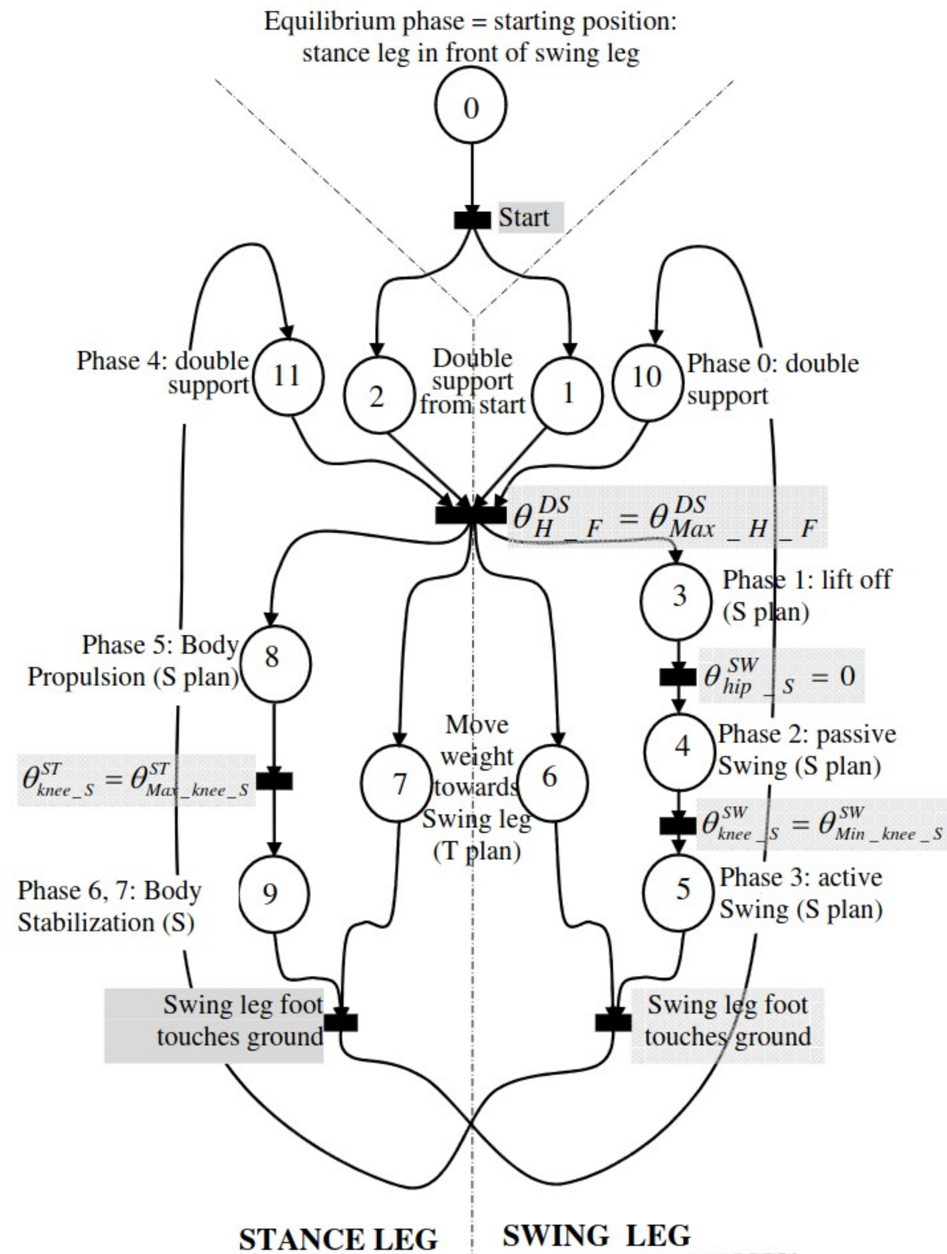
Time domain



Walking Cycle

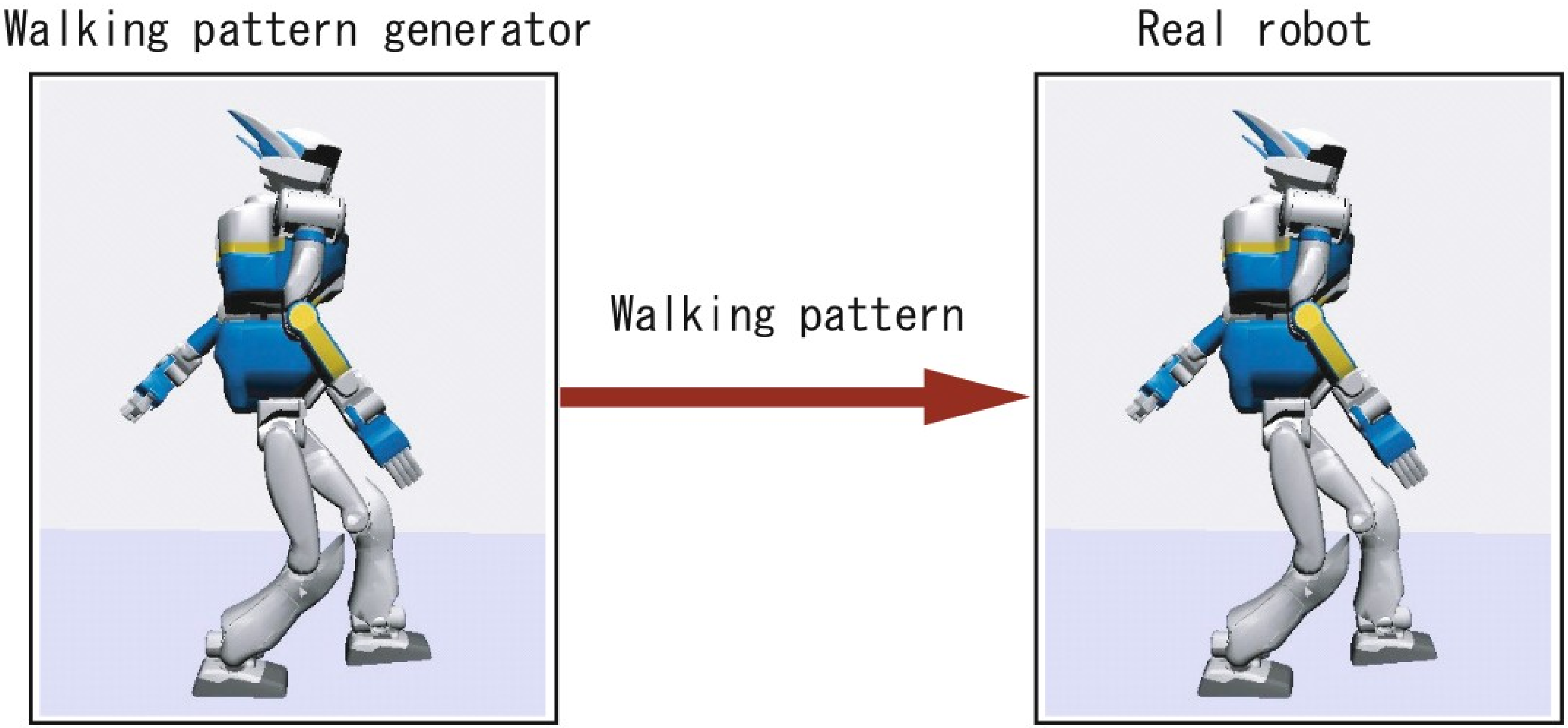


Walking Cycle



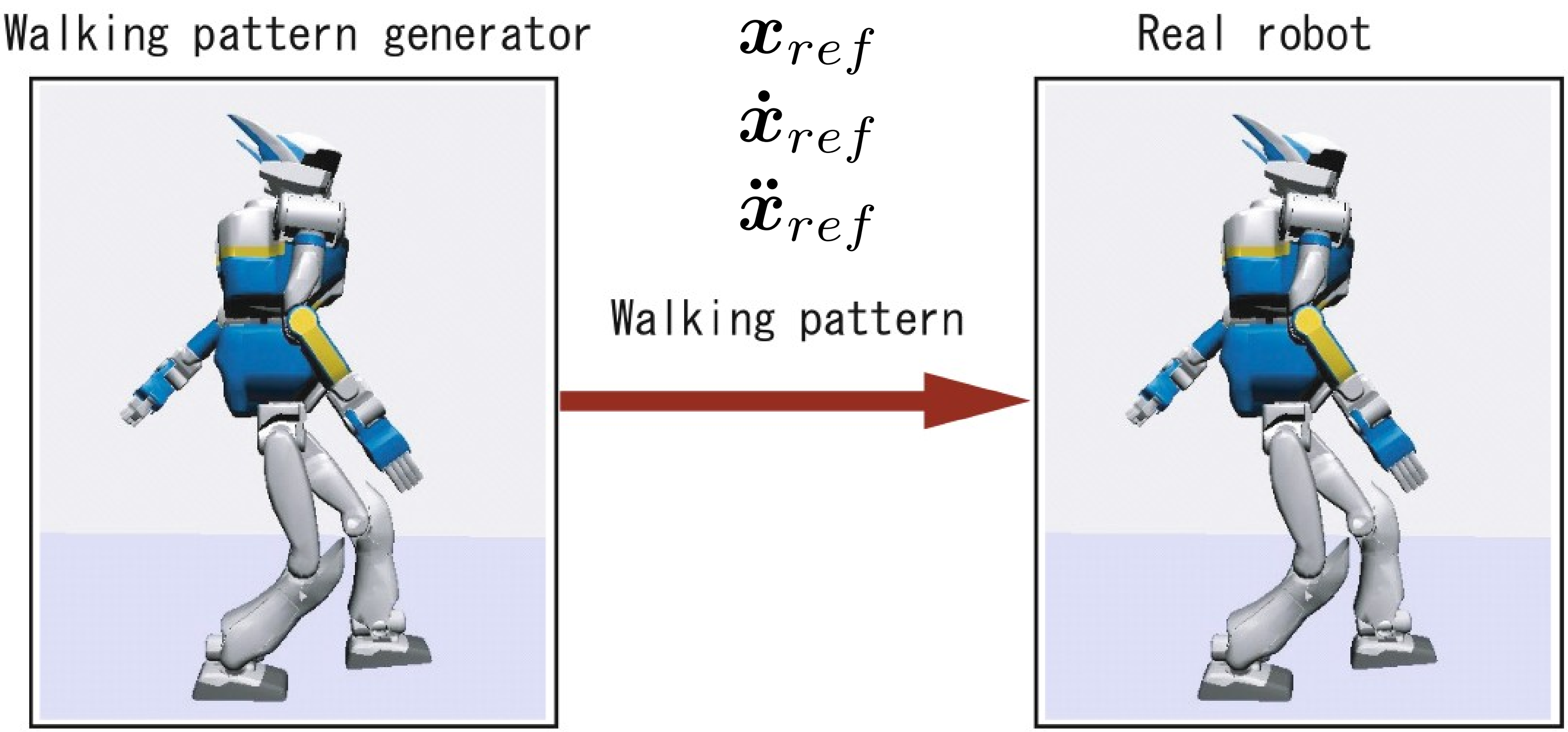
[Serhan et al., 2008]

Walking Motion Generation

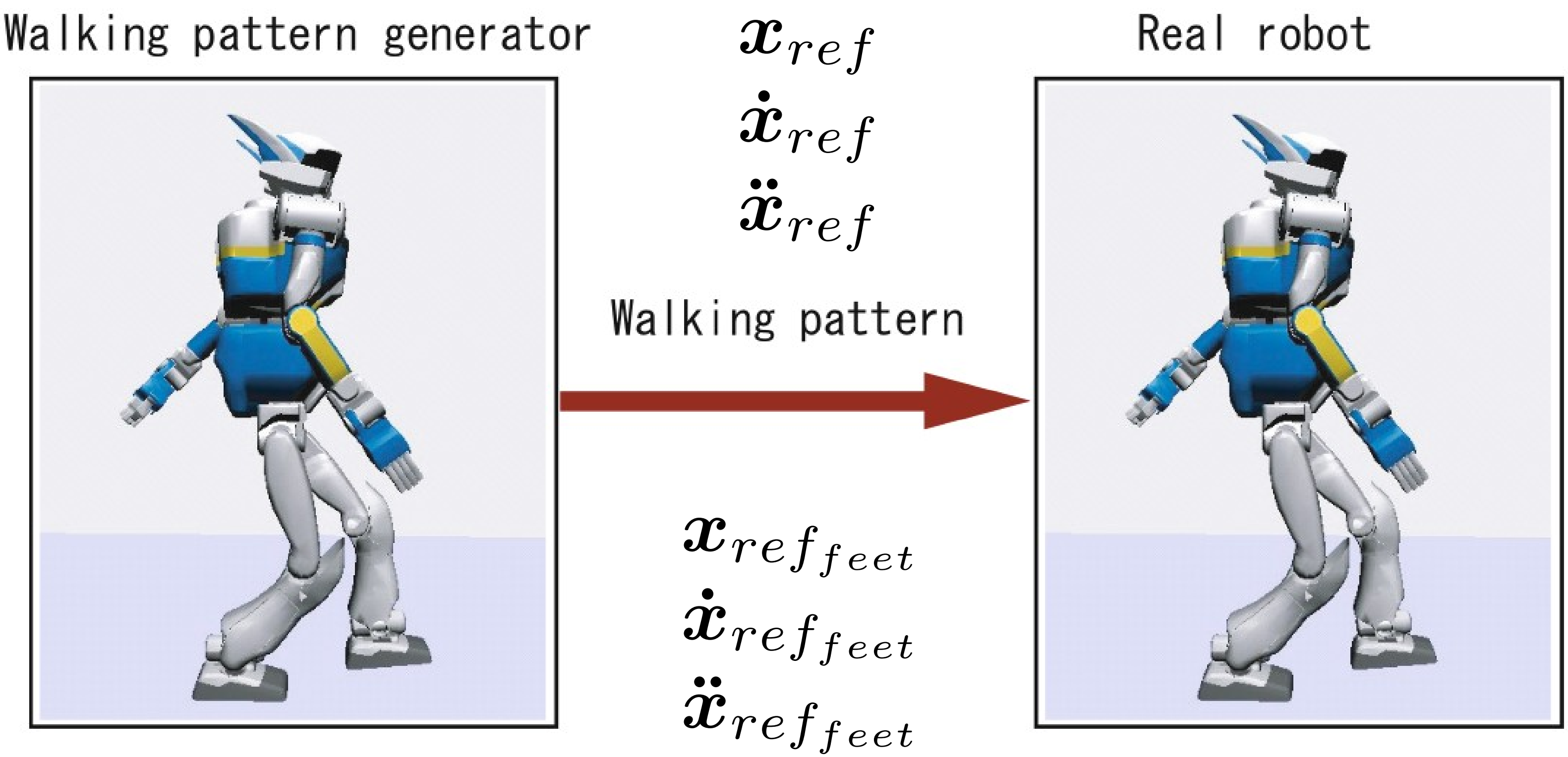


[Kajita et al., *Introduction to humanoid robotics*, Springer 2014]

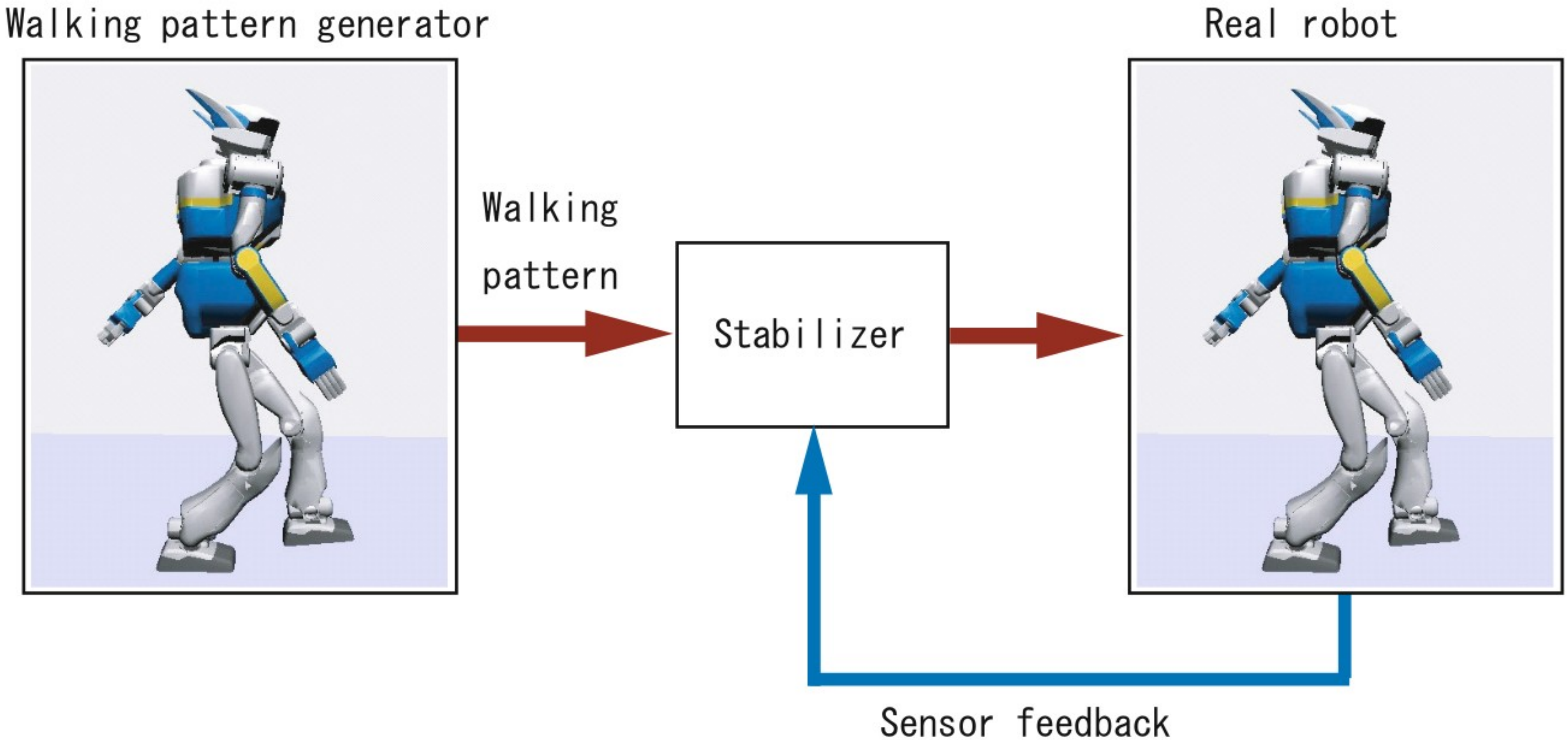
Walking Motion Generation



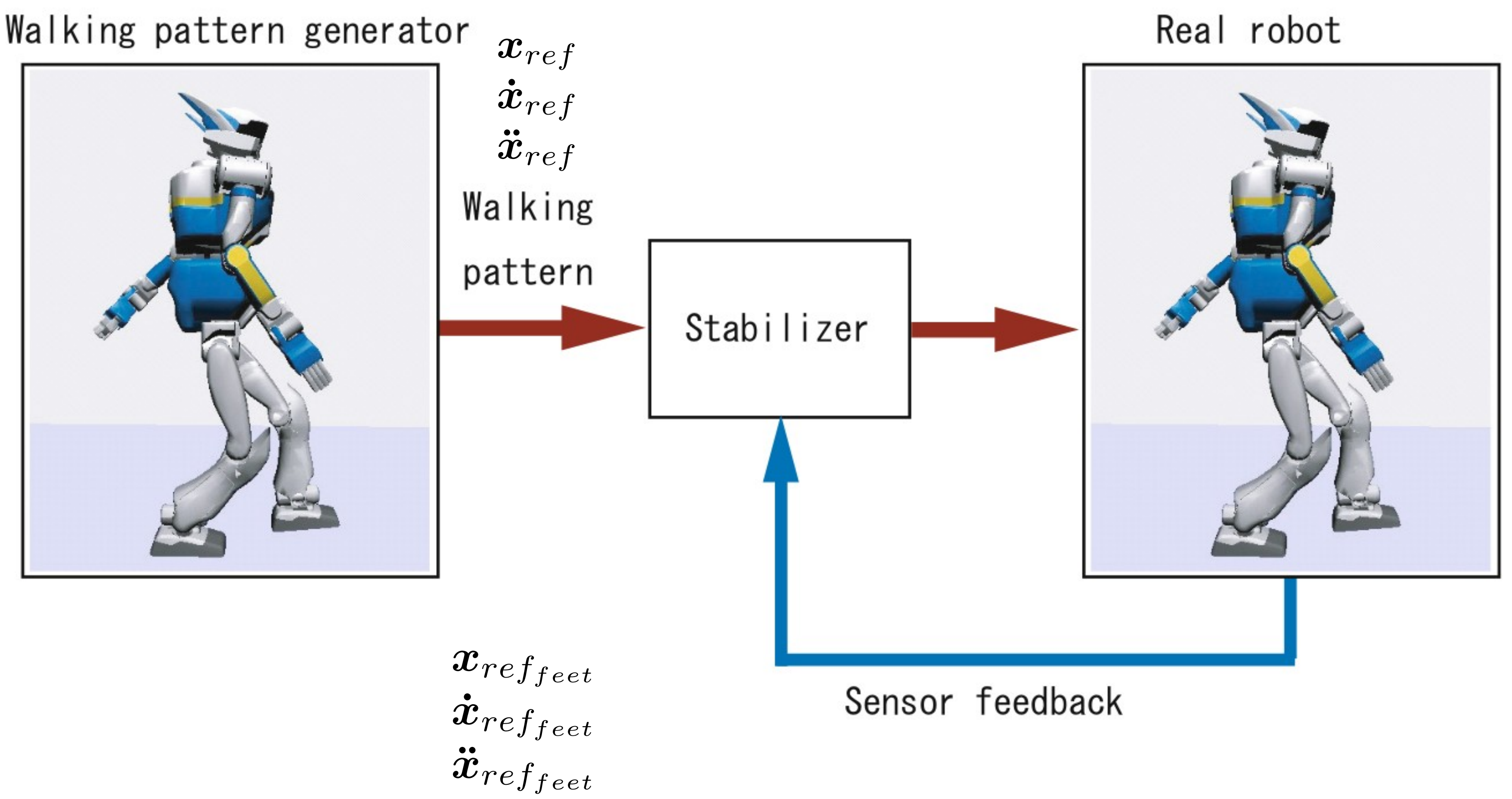
Walking Motion Generation



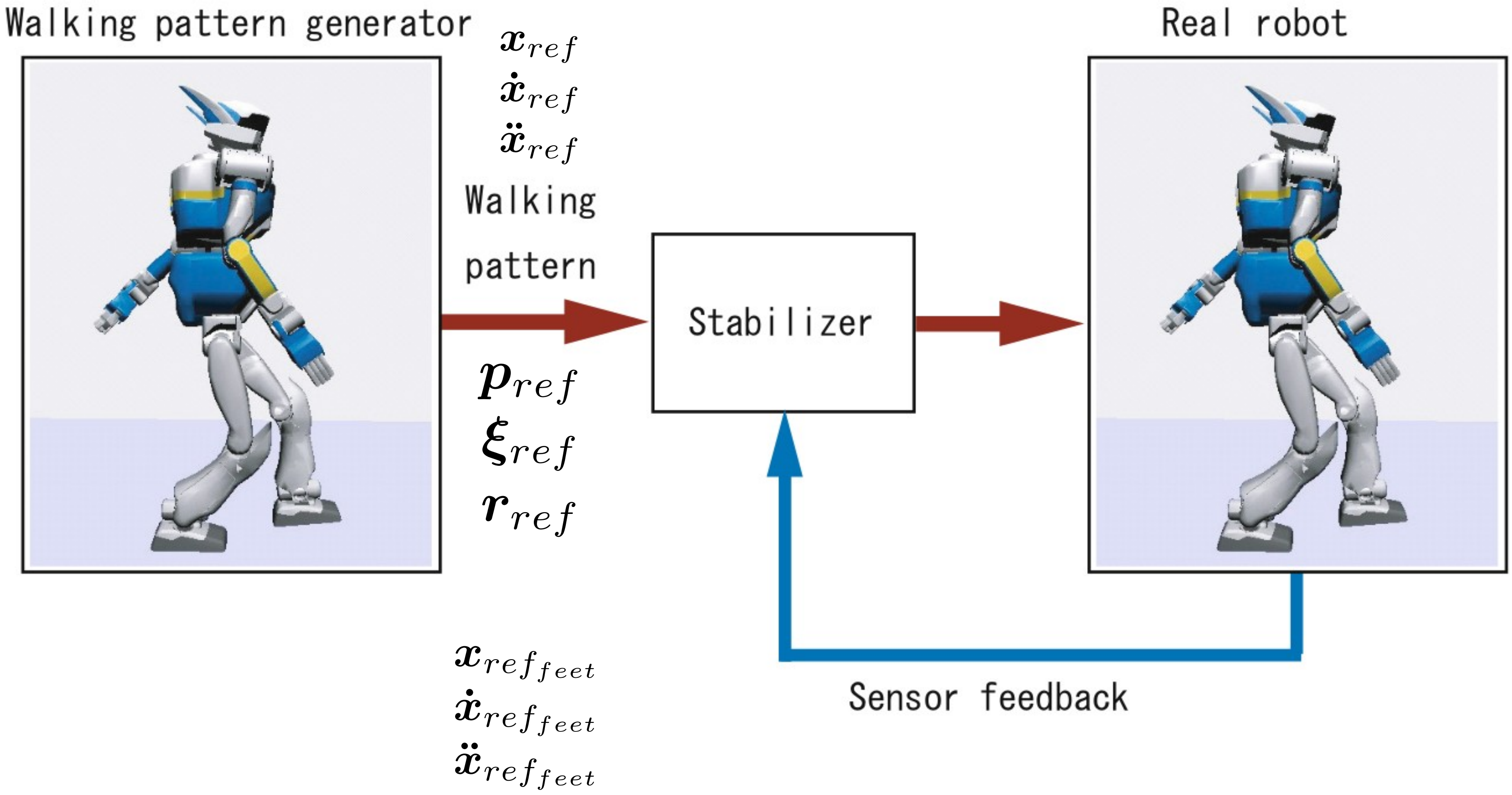
Walking Motion Generation



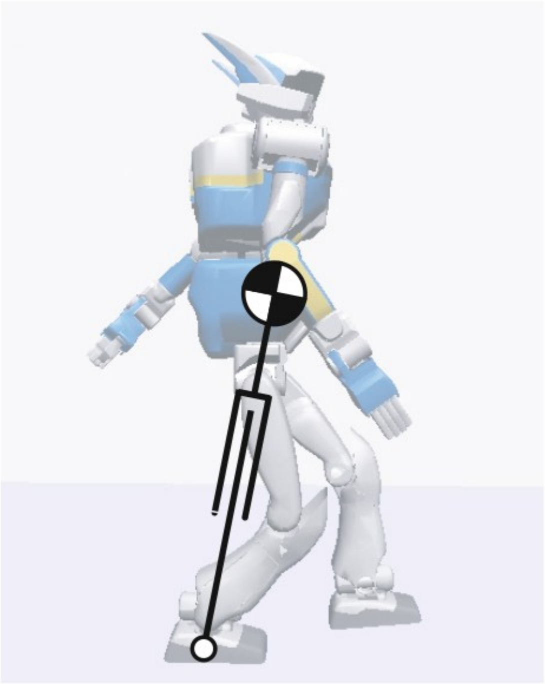
Walking Motion Generation

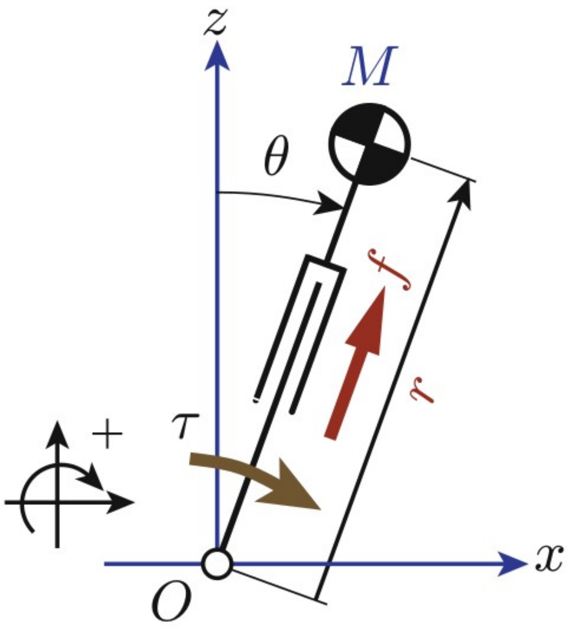


Walking Motion Generation



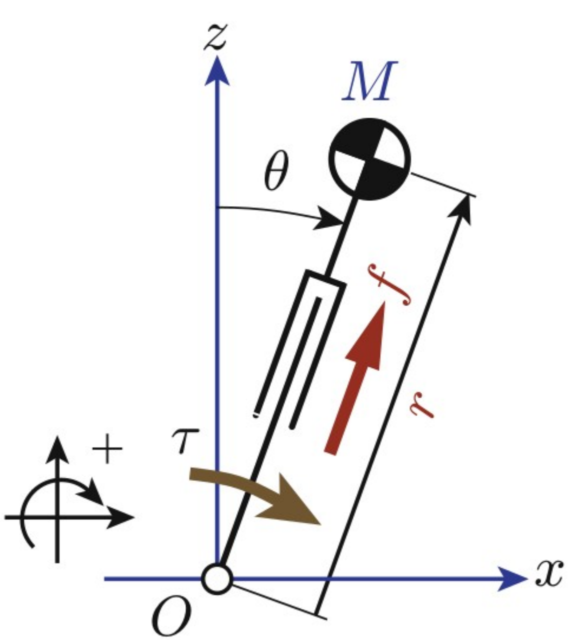
Linear Inverted Pendulum Model

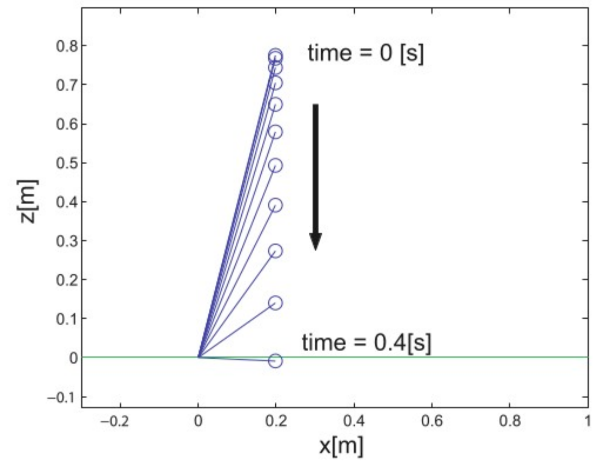




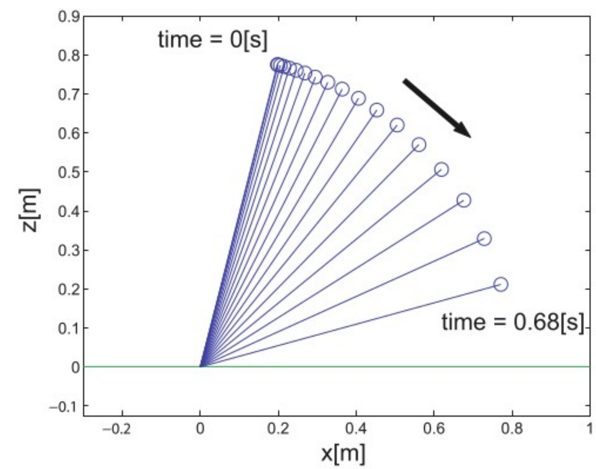
$$\ddot{\boldsymbol{\theta}} = \frac{g}{r} sin(\boldsymbol{\theta})$$

Linear Inverted Pendulum Model

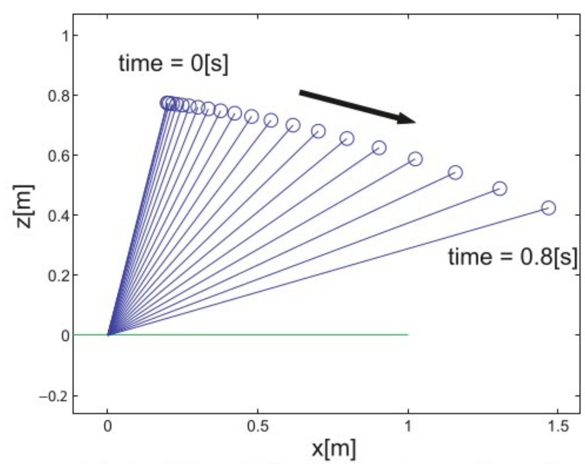




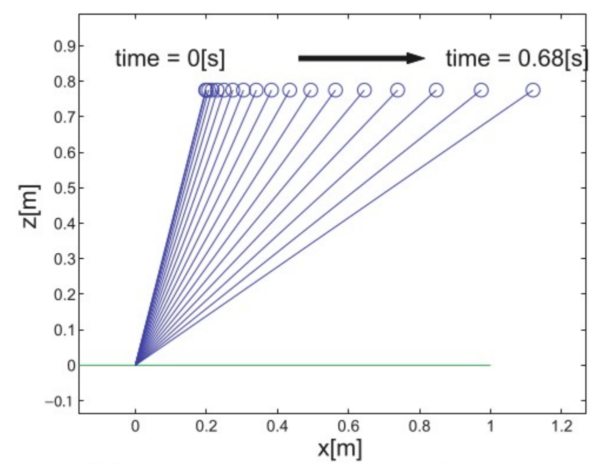
(a) $f = 0$: Free fall of CoM



(b) $f = Mg\cos\theta - Mr\dot{\theta}^2$: Fall down with constant leg length

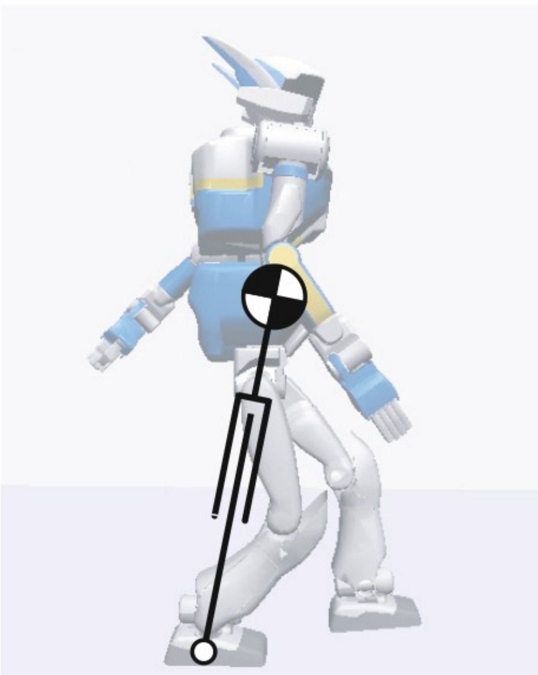


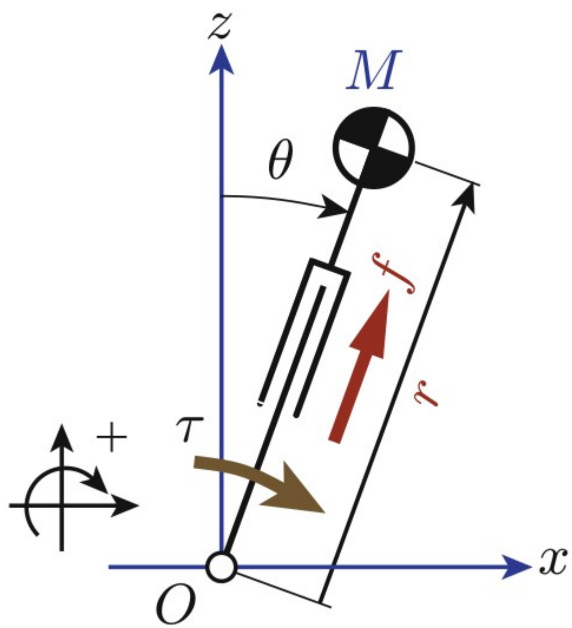
(c) $f = Mg$: Fall down and acceleration



(d) $f = Mg/\cos\theta$: CoM accelerates while keeping the initial height

Linear Inverted Pendulum Model

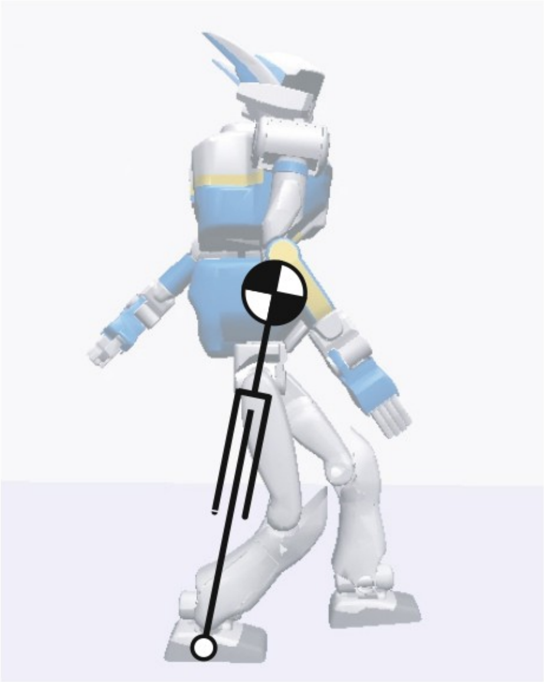


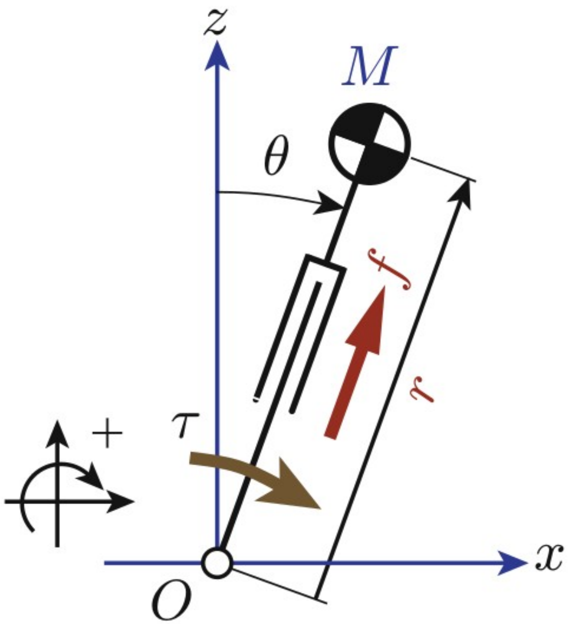


$$\ddot{\boldsymbol{\theta}} = \frac{g}{r} sin(\boldsymbol{\theta})$$

$$\boldsymbol{f} = \frac{Mg}{cos(\boldsymbol{\theta})}$$

Linear Inverted Pendulum Model



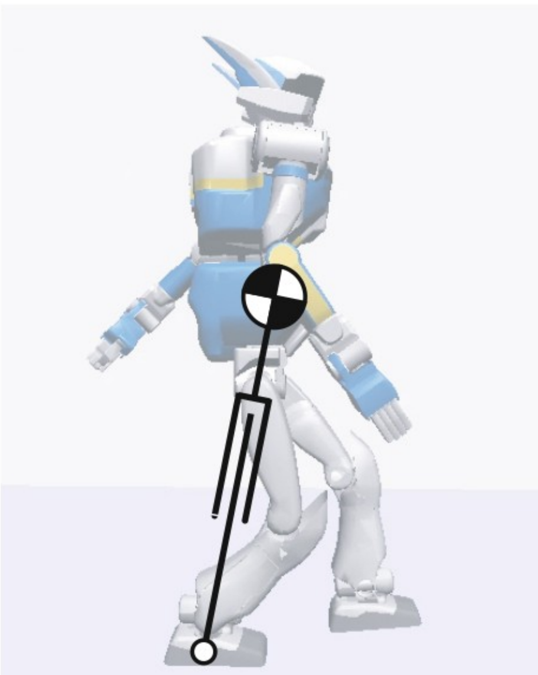


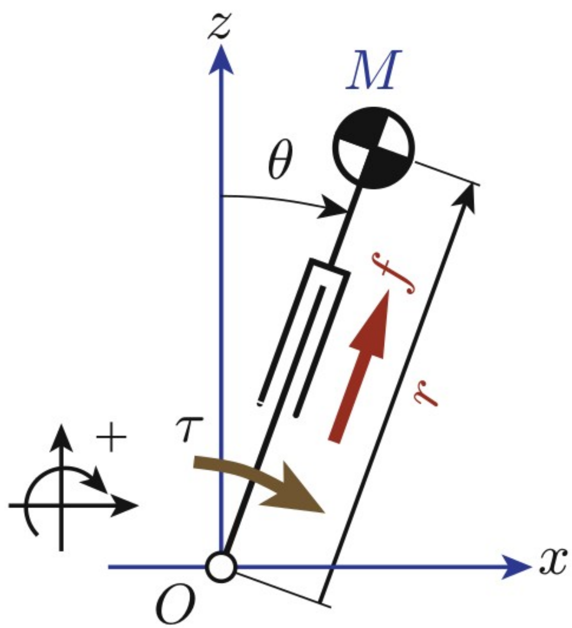
$$\ddot{\boldsymbol{\theta}} = \frac{g}{r} sin(\boldsymbol{\theta})$$

$$\boldsymbol{f} = \frac{Mg}{cos(\boldsymbol{\theta})}$$

$$\ddot{\boldsymbol{x}} = \frac{g}{r} \boldsymbol{x}$$

Linear Inverted Pendulum Model





$$\ddot{\boldsymbol{\theta}} = \frac{g}{r} sin(\boldsymbol{\theta})$$

$$\boldsymbol{f} = \frac{Mg}{cos(\boldsymbol{\theta})}$$

$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model

$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x}$$

Linear Inverted Pendulum Model

$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x}$$

$$\boldsymbol{x}(t) = \boldsymbol{x}(0)\cosh\left(\frac{t}{T_c}\right) + T_c\dot{\boldsymbol{x}}(0)\sinh\left(\frac{t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}(t) = \frac{\boldsymbol{x}(0)}{T_c}\sinh\left(\frac{t}{T_c}\right) + \dot{\boldsymbol{x}}(0)\cosh\left(\frac{t}{T_c}\right)$$

Linear Inverted Pendulum Model

$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x}$$

$$\boldsymbol{x}(t) = \boldsymbol{x}(0)\cosh\left(\frac{t}{T_c}\right) + T_c\dot{\boldsymbol{x}}(0)\sinh\left(\frac{t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}(t) = \frac{\boldsymbol{x}(0)}{T_c}\sinh\left(\frac{t}{T_c}\right) + \dot{\boldsymbol{x}}(0)\cosh\left(\frac{t}{T_c}\right)$$

$$T_c = \sqrt{\frac{z}{g}}$$

Linear Inverted Pendulum Model

$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x}$$

$$\boldsymbol{x}(t) = \boldsymbol{x}(0)\cosh\left(\frac{t}{T_c}\right) + T_c\dot{\boldsymbol{x}}(0)\sinh\left(\frac{t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}(t) = \frac{\boldsymbol{x}(0)}{T_c}\sinh\left(\frac{t}{T_c}\right) + \dot{\boldsymbol{x}}(0)\cosh\left(\frac{t}{T_c}\right)$$

$$T_c = \sqrt{\frac{z}{g}} \qquad \omega_c = \sqrt{\frac{g}{z}}$$

Pendulum Natural Frequency

Linear Inverted Pendulum Model

$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x}$$

$$\boldsymbol{x}(t) = \boldsymbol{x}(0)\cosh\left(\frac{t}{T_c}\right) + T_c\dot{\boldsymbol{x}}(0)\sinh\left(\frac{t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}(t) = \frac{\boldsymbol{x}(0)}{T_c}\sinh\left(\frac{t}{T_c}\right) + \dot{\boldsymbol{x}}(0)\cosh\left(\frac{t}{T_c}\right)$$

$$T_c = \sqrt{\frac{z}{g}} \qquad \omega_c = \sqrt{\frac{g}{z}}$$

Pendulum Natural Frequency

$$f_c = \frac{1}{\pi}\sqrt{\frac{g}{z}}$$

Linear Inverted Pendulum Model

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 \cosh\left(\frac{\Delta t}{T_c}\right) + T_c \dot{\boldsymbol{x}}_0 \sinh\left(\frac{\Delta t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0}{T_c} \sinh\left(\frac{\Delta t}{T_c}\right) + \dot{\boldsymbol{x}}_0 \cosh\left(\frac{\Delta t}{T_c}\right)$$

Linear Inverted Pendulum Model

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 \cosh\left(\frac{\Delta t}{T_c}\right) + T_c \dot{\boldsymbol{x}}_0 \sinh\left(\frac{\Delta t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0}{T_c} \sinh\left(\frac{\Delta t}{T_c}\right) + \dot{\boldsymbol{x}}_0 \cosh\left(\frac{\Delta t}{T_c}\right)$$

$$\cosh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} + e^{-\boldsymbol{x}}}{2} \qquad \sinh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} - e^{-\boldsymbol{x}}}{2}$$

Linear Inverted Pendulum Model

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 \cosh\left(\frac{\Delta t}{T_c}\right) + T_c \dot{\boldsymbol{x}}_0 \sinh\left(\frac{\Delta t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0}{T_c} \sinh\left(\frac{\Delta t}{T_c}\right) + \dot{\boldsymbol{x}}_0 \cosh\left(\frac{\Delta t}{T_c}\right)$$

$$\cosh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} + e^{-\boldsymbol{x}}}{2} \qquad \sinh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} - e^{-\boldsymbol{x}}}{2}$$

$$\boldsymbol{x}_1 = \frac{\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0}{2} e^{\frac{\Delta t}{T_c}} + \frac{\boldsymbol{x}_0 - T_c \dot{\boldsymbol{x}}_0}{2} e^{-\frac{\Delta t}{T_c}}$$

$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0}{2T_c} e^{\frac{\Delta t}{T_c}} - \frac{\boldsymbol{x}_0 - T_c \dot{\boldsymbol{x}}_0}{2T_c} e^{-\frac{\Delta t}{T_c}}$$

Linear Inverted Pendulum Model

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 \cosh\left(\frac{\Delta t}{T_c}\right) + T_c \dot{\boldsymbol{x}}_0 \sinh\left(\frac{\Delta t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0}{T_c} \sinh\left(\frac{\Delta t}{T_c}\right) + \dot{\boldsymbol{x}}_0 \cosh\left(\frac{\Delta t}{T_c}\right)$$

$$\cosh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} + e^{-\boldsymbol{x}}}{2} \qquad \sinh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} - e^{-\boldsymbol{x}}}{2}$$

$$\boldsymbol{x}_1 = \frac{\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0}{2} e^{\frac{\Delta t}{T_c}} + \frac{\boldsymbol{x}_0 - T_c \dot{\boldsymbol{x}}_0}{2} e^{-\frac{\Delta t}{T_c}}$$

$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0}{2T_c} e^{\frac{\Delta t}{T_c}} - \frac{\boldsymbol{x}_0 - T_c \dot{\boldsymbol{x}}_0}{2T_c} e^{-\frac{\Delta t}{T_c}}$$

$$\boldsymbol{x}_1 + T_c \dot{\boldsymbol{x}}_1 = \left(\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0\right) e^{\frac{\Delta t}{T_c}}$$

Linear Inverted Pendulum Model

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 \cosh\left(\frac{\Delta t}{T_c}\right) + T_c \dot{\boldsymbol{x}}_0 \sinh\left(\frac{\Delta t}{T_c}\right)$$

$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0}{T_c} \sinh\left(\frac{\Delta t}{T_c}\right) + \dot{\boldsymbol{x}}_0 \cosh\left(\frac{\Delta t}{T_c}\right)$$

$$\cosh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} + e^{-\boldsymbol{x}}}{2} \qquad \sinh \boldsymbol{x} = \frac{e^{\boldsymbol{x}} - e^{-\boldsymbol{x}}}{2}$$

$$\boldsymbol{x}_1 = \frac{\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0}{2} e^{\frac{\Delta t}{T_c}} + \frac{\boldsymbol{x}_0 - T_c \dot{\boldsymbol{x}}_0}{2} e^{-\frac{\Delta t}{T_c}}$$

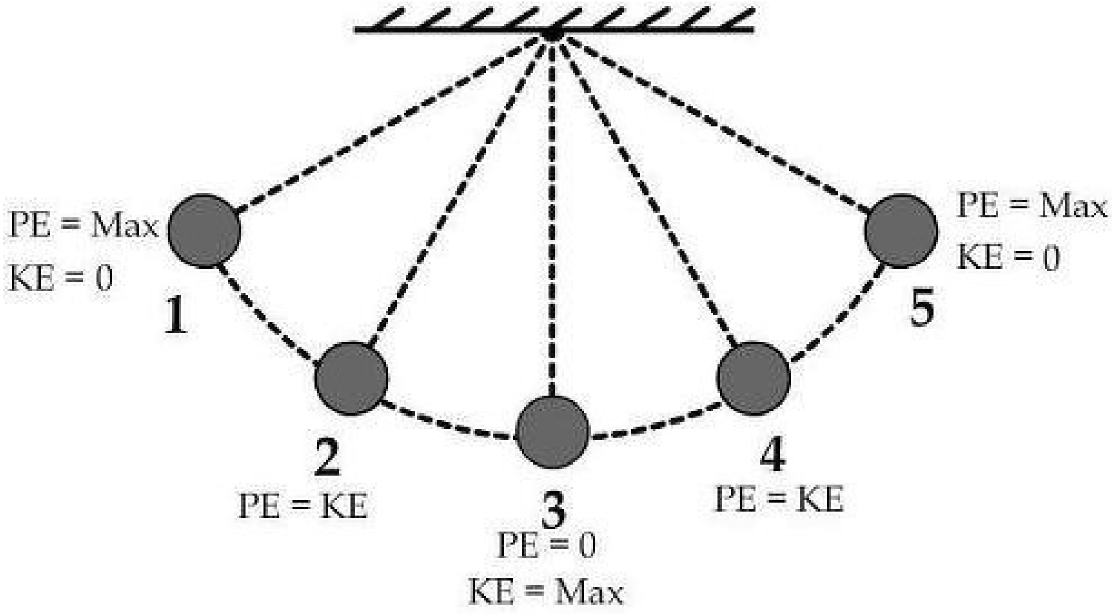
$$\dot{\boldsymbol{x}}_1 = \frac{\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0}{2T_c} e^{\frac{\Delta t}{T_c}} - \frac{\boldsymbol{x}_0 - T_c \dot{\boldsymbol{x}}_0}{2T_c} e^{-\frac{\Delta t}{T_c}}$$

$$\boldsymbol{x}_1 + T_c \dot{\boldsymbol{x}}_1 = \left(\boldsymbol{x}_0 + T_c \dot{\boldsymbol{x}}_0\right) e^{\frac{\Delta t}{T_c}}$$

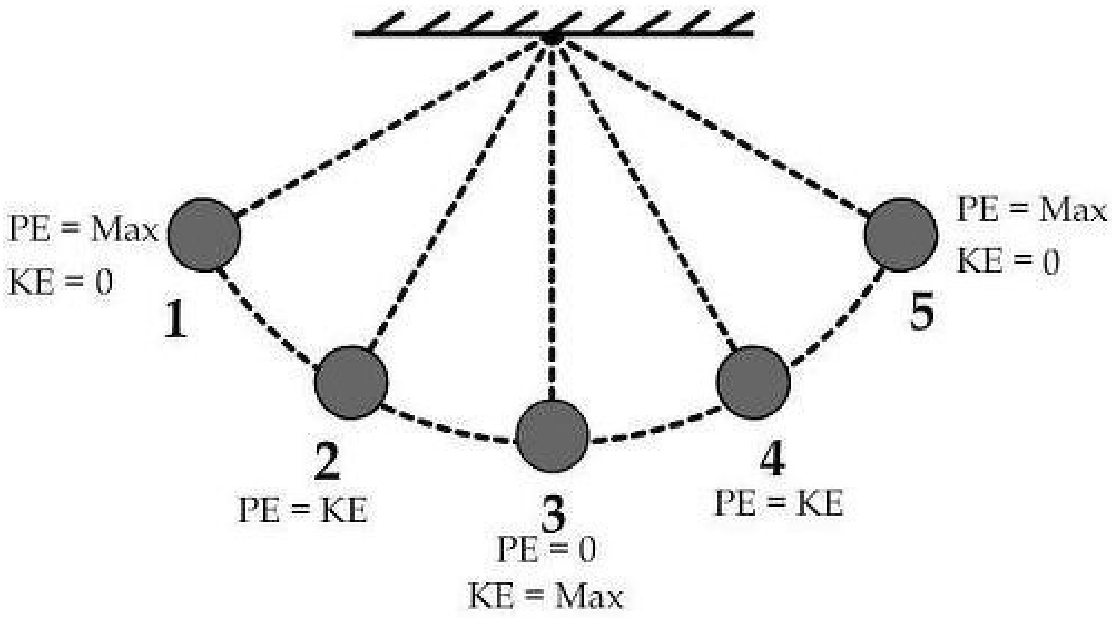
$$\Delta t = T_c \ln \frac{\boldsymbol{x}_0 - T_c \dot{\boldsymbol{x}}_0}{\boldsymbol{x}_1 - T_c \dot{\boldsymbol{x}}_1}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model



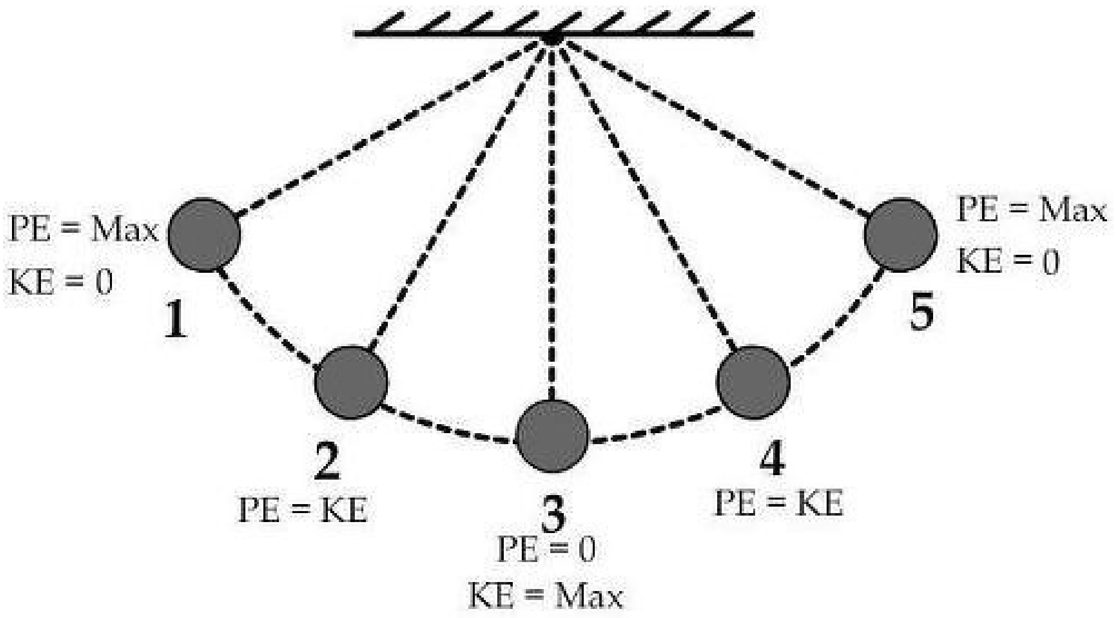
Linear Inverted Pendulum Model



$$KE_{max} = \frac{1}{2} M \dot{\boldsymbol{x}}_{max}$$

$$PE_{max} = M g z_{max}$$

Linear Inverted Pendulum Model



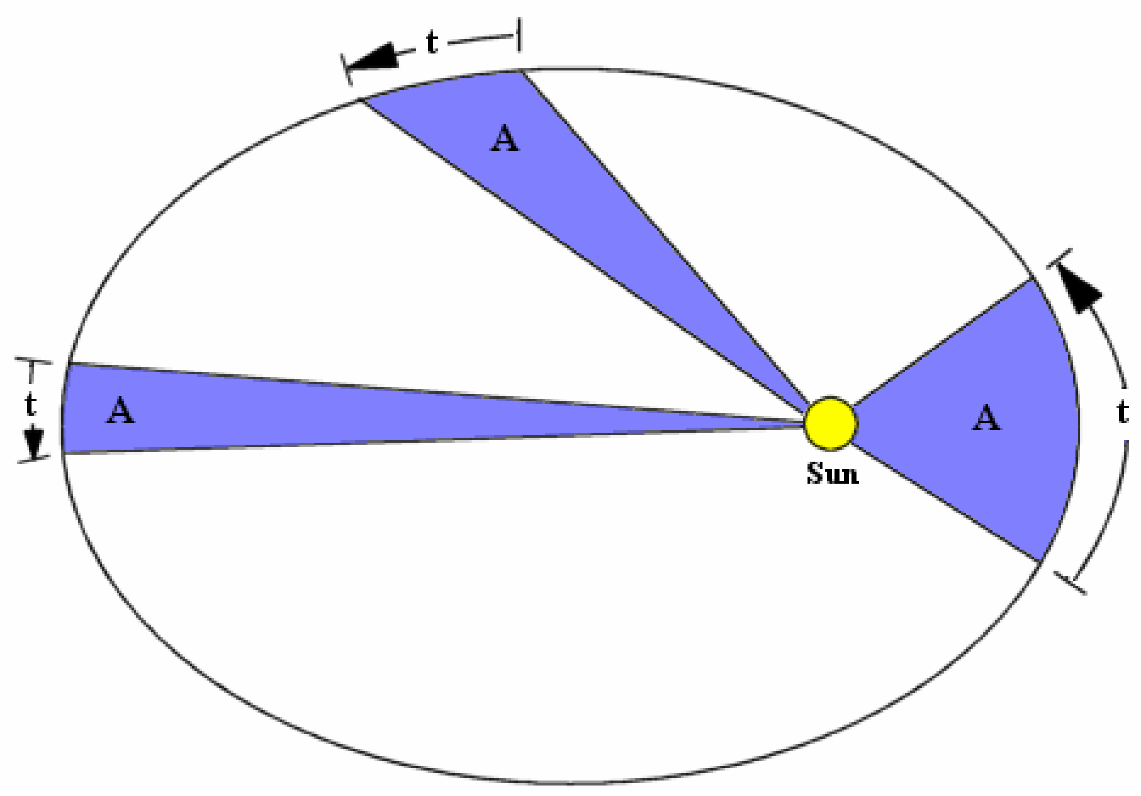
$$KE_{max} = \frac{1}{2} M \dot{\boldsymbol{x}}_{max}$$

$$PE_{max} = M g z_{max}$$

$$KE_{max} = PE_{max}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model



$$KE_{max} = \frac{1}{2} M \dot{\boldsymbol{x}}_{max}$$

$$PE_{max} = M g z_{max}$$

$$KE_{max} = PE_{max}$$

Linear Inverted Pendulum Model

$$\dot{\boldsymbol{x}}\left(\ddot{\boldsymbol{x}} - \frac{g}{z}\boldsymbol{x}\right) = 0$$

$$\int \{\ddot{x}\dot{x} - \frac{g}{z}x\dot{x}\}dt = constant$$

$$\frac{1}{2}\dot{\boldsymbol{x}}^2 - \frac{g}{2z}\boldsymbol{x}^2 = E$$

KE PE

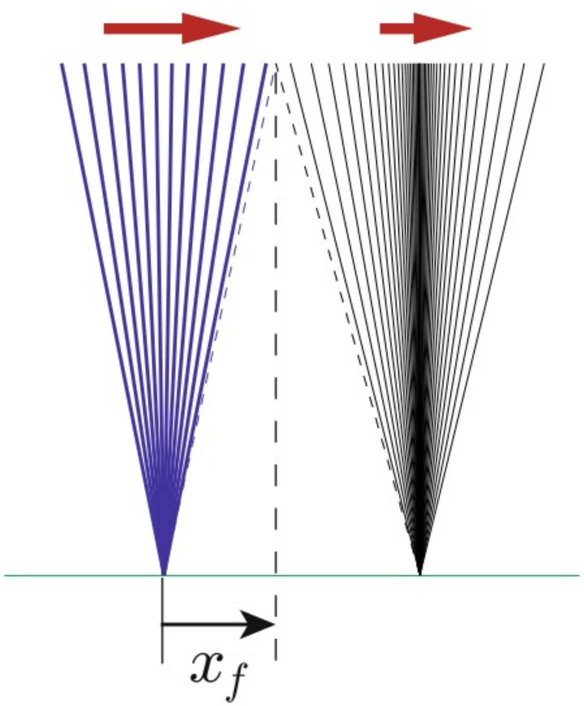
Linear Inverted Pendulum Model

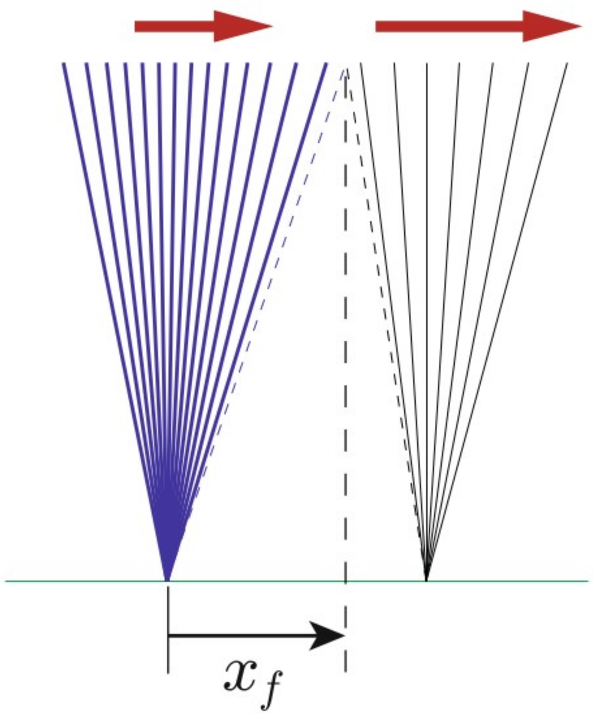
$$\dot{\boldsymbol{x}}\left(\ddot{\boldsymbol{x}}-\frac{g}{z}\boldsymbol{x}\right)=0$$

$$\int\{\ddot{x}\dot{x}-\frac{g}{z}x\dot{x}\}dt=constant$$

$$\frac{1}{2}\dot{\boldsymbol{x}}^2-\frac{g}{2z}\boldsymbol{x}^2=E$$

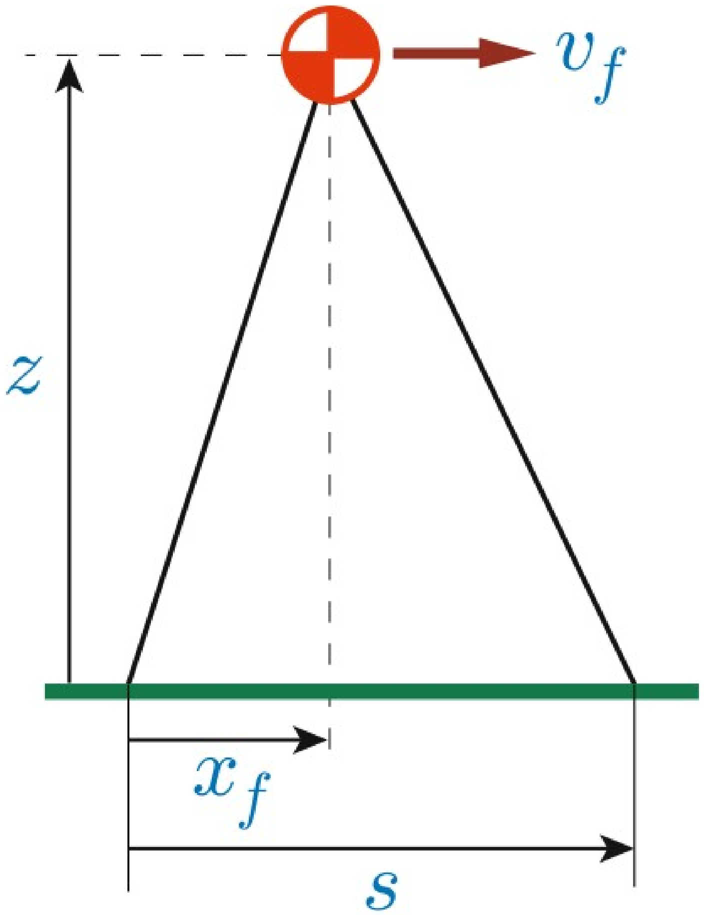
KE PE



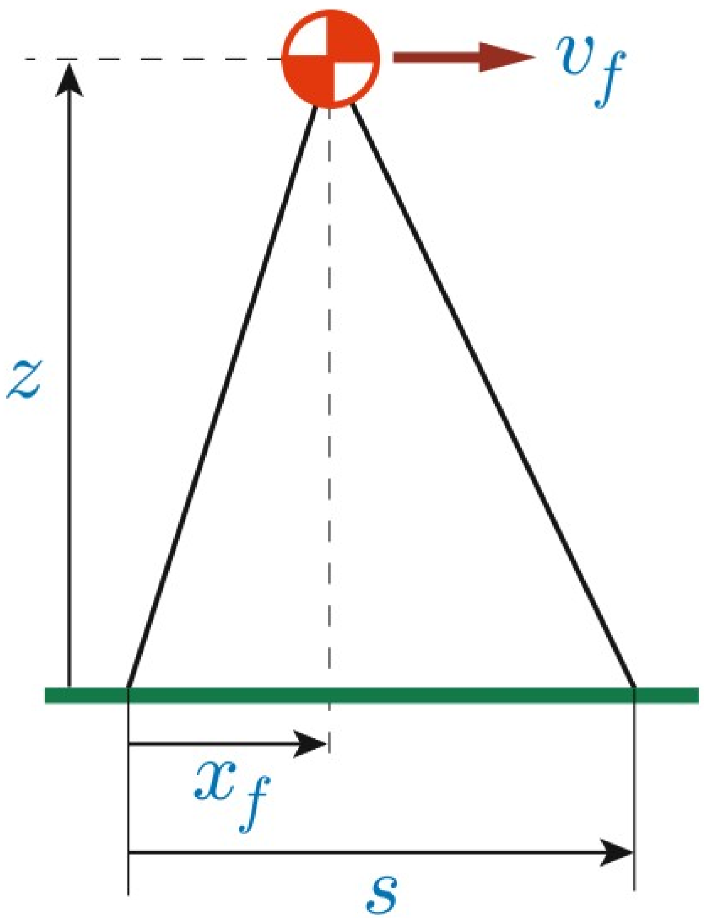


Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model



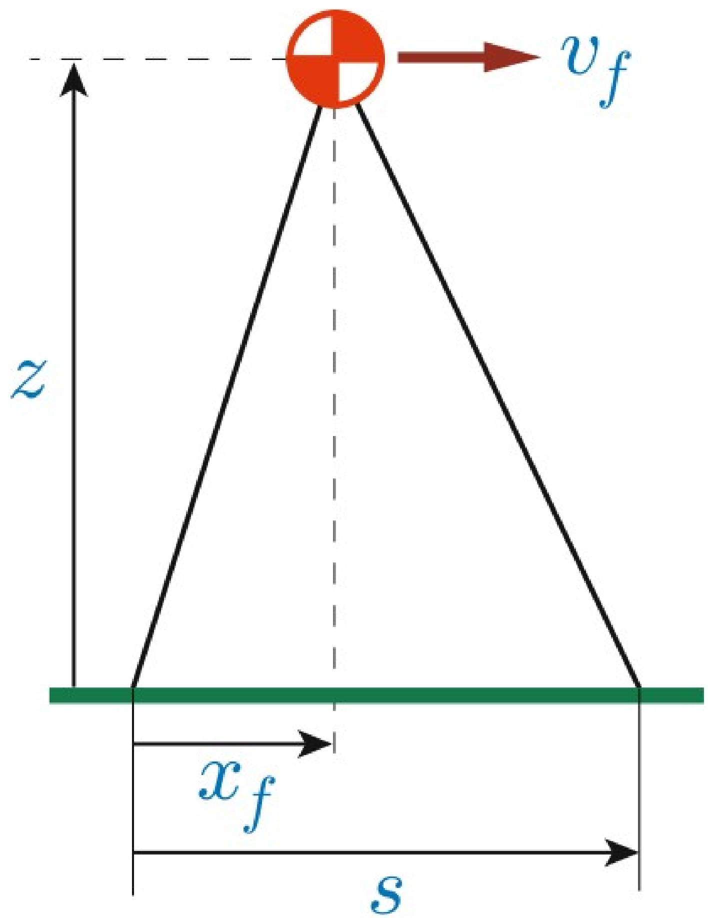
Linear Inverted Pendulum Model



$$E_1 = \frac{1}{2}\boldsymbol{v}_f^2 - \frac{g}{2z}\boldsymbol{x}_f^2$$

$$E_2 = \frac{1}{2}\boldsymbol{v}_f^2 - \frac{g}{2z}\left(\boldsymbol{x}_f - s\right)^2$$

Linear Inverted Pendulum Model



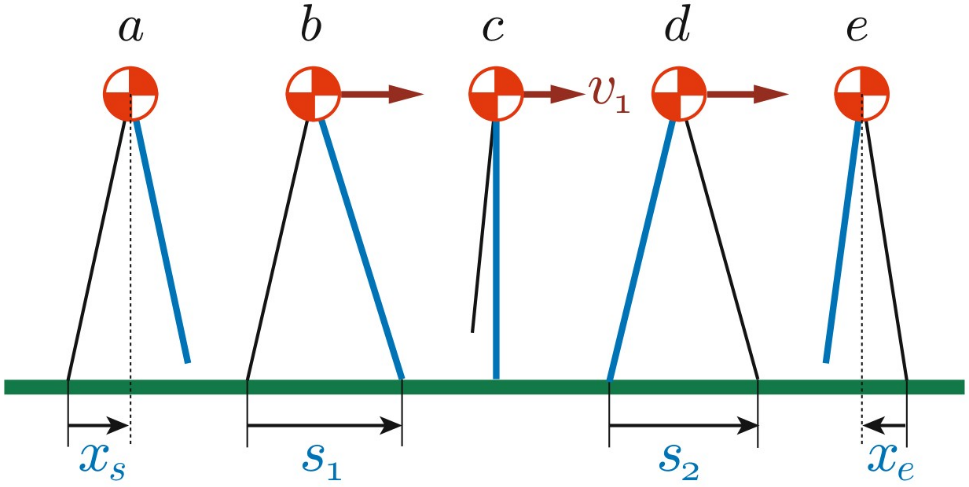
$$E_1 = \frac{1}{2}\boldsymbol{v}_f^2 - \frac{g}{2z}\boldsymbol{x}_f^2$$

$$E_2 = \frac{1}{2}\boldsymbol{v}_f^2 - \frac{g}{2z}\left(\boldsymbol{x}_f - s\right)^2$$

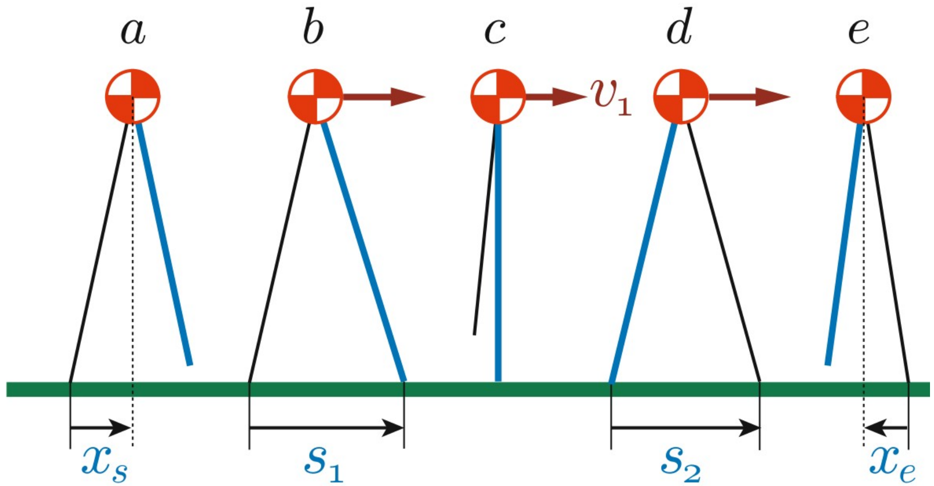
$$\boldsymbol{x}_f = \frac{z}{gs}\left(E_2 - E_1\right) + \frac{s}{2}$$

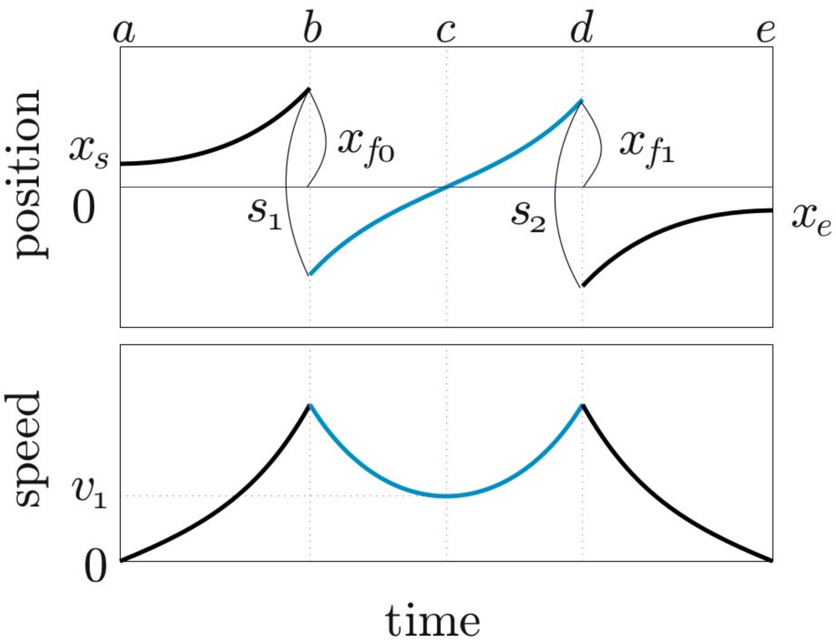
$$\boldsymbol{v}_f = \sqrt{2E_1 + \frac{g}{z}\boldsymbol{x}_f^2}$$

Linear Inverted Pendulum Model



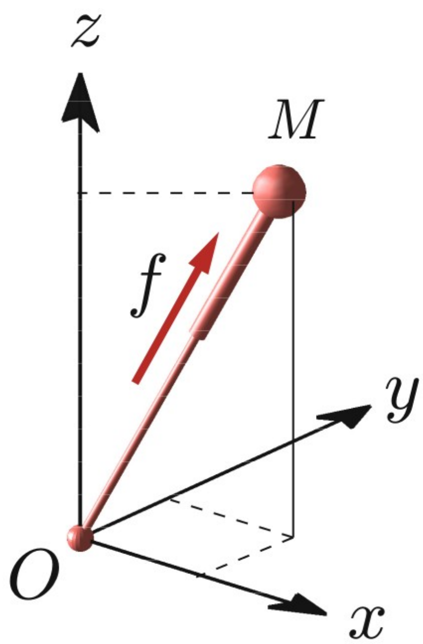
Linear Inverted Pendulum Model





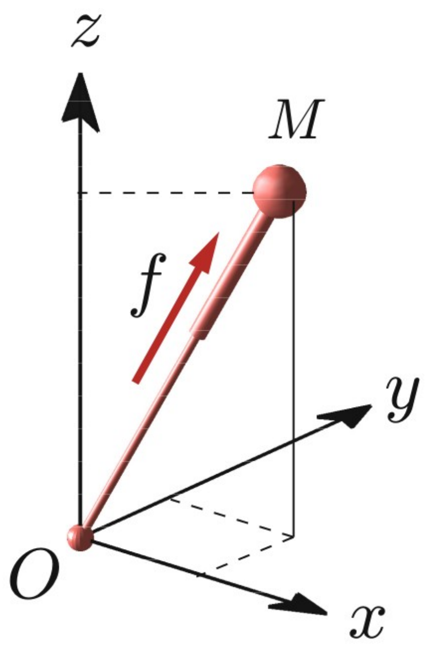
Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model 3D



Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

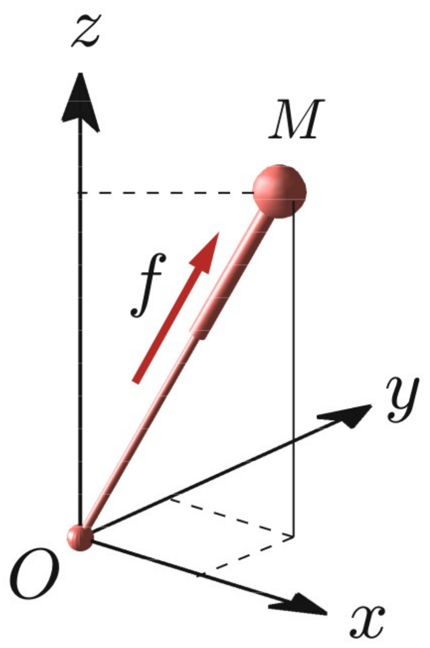
Linear Inverted Pendulum Model 3D

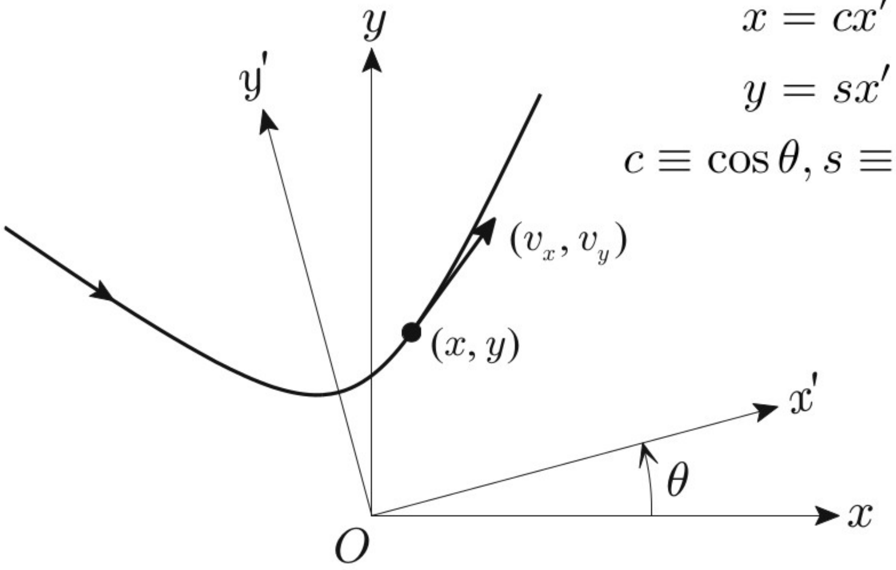


$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x} \quad \ddot{\boldsymbol{y}} = \frac{g}{z}\boldsymbol{y}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model 3D



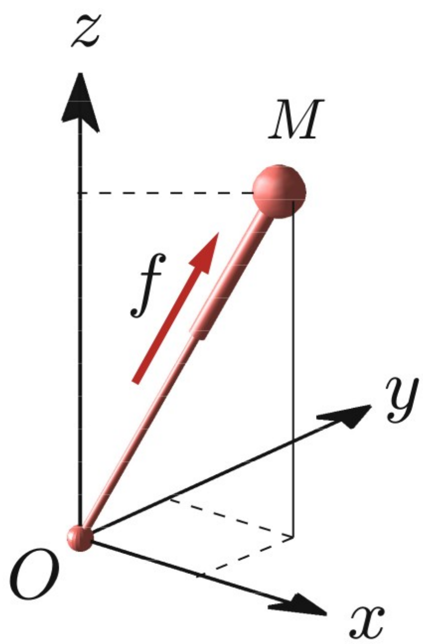


$$x = cx' - sy'$$
$$y = sx' + cy'$$
$$c \equiv \cos\theta, s \equiv \sin\theta.$$

$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x} \quad \ddot{\boldsymbol{y}} = \frac{g}{z}\boldsymbol{y}$$

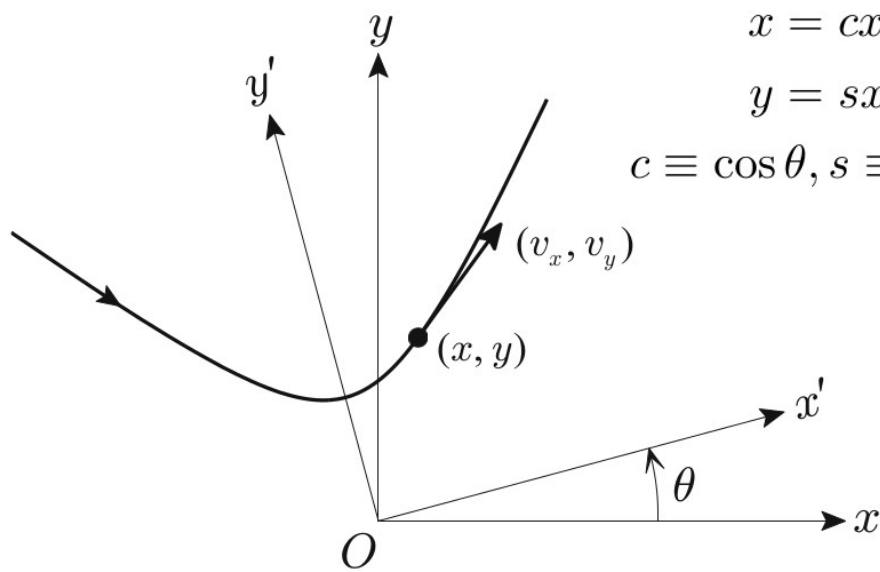
Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model 3D



$$\ddot{\boldsymbol{x}} = \frac{g}{z}\boldsymbol{x} \qquad \ddot{\boldsymbol{y}} = \frac{g}{z}\boldsymbol{y}$$

$$x = cx' - sy'$$
$$y = sx' + cy'$$
$$c \equiv \cos\theta, s \equiv \sin\theta.$$

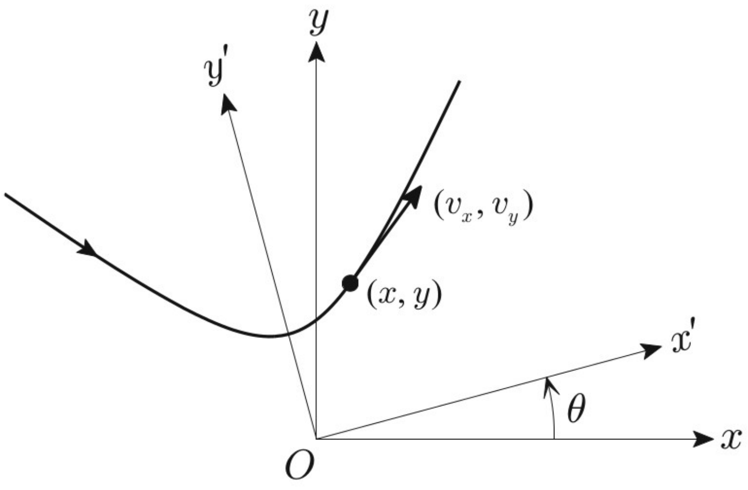


$$\ddot{\boldsymbol{x}}' = \frac{g}{z}\boldsymbol{x}' \qquad \ddot{\boldsymbol{y}}' = \frac{g}{z}\boldsymbol{y}'$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Linear Inverted Pendulum Model 3D

$$E_x' = -\frac{g}{2z}\left(c\boldsymbol{x} + s\boldsymbol{y}\right)^2 + \frac{1}{2}\left(c\dot{\boldsymbol{x}} + s\dot{\boldsymbol{y}}\right)$$



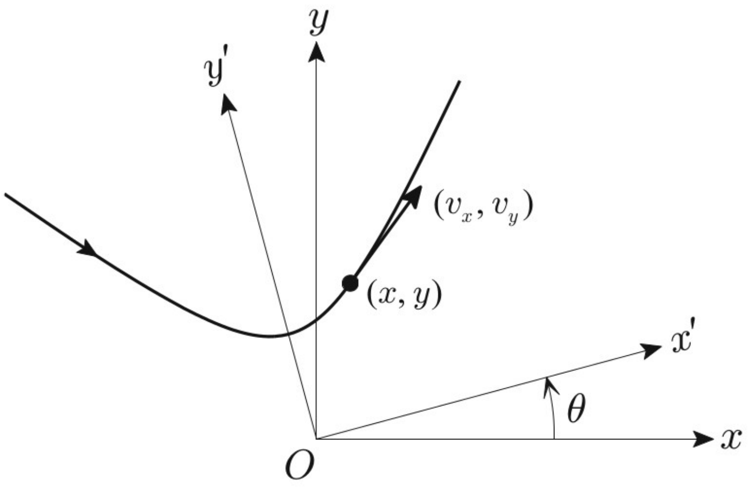
Linear Inverted Pendulum Model 3D

$$E'_x = -\frac{g}{2z}\left(c\boldsymbol{x} + s\boldsymbol{y}\right)^2 + \frac{1}{2}\left(c\dot{\boldsymbol{x}} + s\dot{\boldsymbol{y}}\right)$$

$$\frac{\partial E'_x}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta$$

$$A = \frac{g}{z}\boldsymbol{x}\boldsymbol{y} - \dot{\boldsymbol{x}}\dot{\boldsymbol{y}}$$

$$B = \frac{g}{z}\left(\boldsymbol{x}^2 - \boldsymbol{y}^2\right) - \left(\dot{\boldsymbol{x}}^2 - \dot{\boldsymbol{y}}^2\right)$$



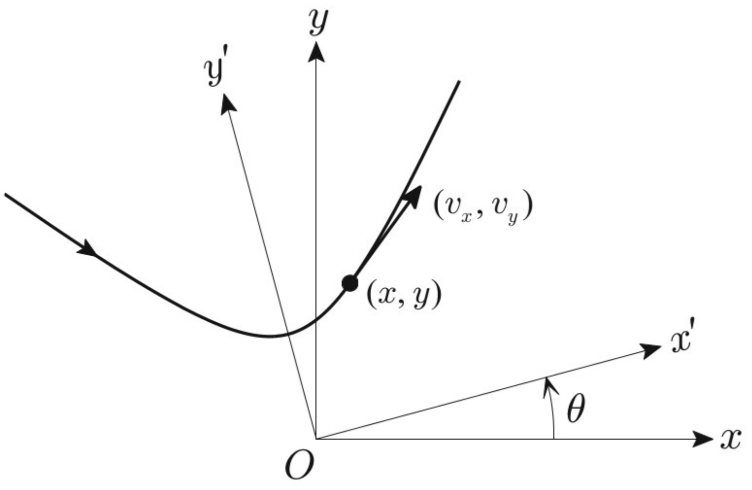
Linear Inverted Pendulum Model 3D

$$E'_x = -\frac{g}{2z}\left(c\boldsymbol{x} + s\boldsymbol{y}\right)^2 + \frac{1}{2}\left(c\dot{\boldsymbol{x}} + s\dot{\boldsymbol{y}}\right)$$

$$\frac{\partial E'_x}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta$$

$$A = \frac{g}{z}\boldsymbol{x}\boldsymbol{y} - \dot{\boldsymbol{x}}\dot{\boldsymbol{y}}$$

$$B = \frac{g}{z}\left(\boldsymbol{x}^2 - \boldsymbol{y}^2\right) - \left(\dot{\boldsymbol{x}}^2 - \dot{\boldsymbol{y}}^2\right)$$



$$\theta = \begin{cases} (1/2)\tan^{-1}(2A/B) & (\text{if } B \neq 0) \\ \pi/4 & (\text{if } A \neq 0, B = 0) \end{cases}$$

Linear Inverted Pendulum Model 3D

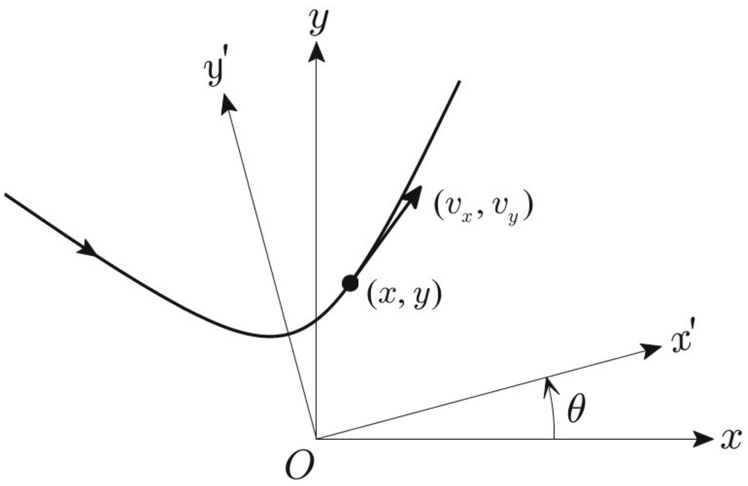
$$E'_x = -\frac{g}{2z}(c\boldsymbol{x} + s\boldsymbol{y})^2 + \frac{1}{2}(c\dot{\boldsymbol{x}} + s\dot{\boldsymbol{y}})$$

$$\frac{\partial E'_x}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta$$

$$A = \frac{g}{z}\boldsymbol{x}\boldsymbol{y} - \dot{\boldsymbol{x}}\dot{\boldsymbol{y}}$$

$$B = \frac{g}{z}\left(\boldsymbol{x}^2 - \boldsymbol{y}^2\right) - \left(\dot{\boldsymbol{x}}^2 - \dot{\boldsymbol{y}}^2\right)$$

$$\theta = \begin{cases} (1/2)\tan^{-1}(2A/B) & (\text{if } B \neq 0) \\ \pi/4 & (\text{if } A \neq 0, B = 0) \end{cases}$$



$$E'_{x_{max}}$$

when

$$\theta = 0$$

Linear Inverted Pendulum Model 3D

$$\frac{\partial E'_x}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta \qquad\qquad \theta = 0$$

Linear Inverted Pendulum Model 3D

$$\frac{\partial E_x'}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta \qquad\qquad \theta = 0$$

$$\frac{g}{z}\boldsymbol{x}\boldsymbol{y} - \dot{\boldsymbol{x}}\dot{\boldsymbol{y}} = 0$$

Linear Inverted Pendulum Model 3D

$$\frac{\partial E'_x}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta \qquad \theta = 0$$

$$\frac{g}{z}\boldsymbol{x}\boldsymbol{y} - \dot{\boldsymbol{x}}\dot{\boldsymbol{y}} = 0$$

$$\dot{\boldsymbol{x}}^2 = 2E_x + \frac{g}{z}\boldsymbol{x}^2 \qquad \dot{\boldsymbol{y}}^2 = 2E_y + \frac{g}{z}\boldsymbol{y}^2$$

Linear Inverted Pendulum Model 3D

$$\frac{\partial E'_x}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta \qquad \theta = 0$$

$$\frac{g}{z}\boldsymbol{x}\boldsymbol{y} - \dot{\boldsymbol{x}}\dot{\boldsymbol{y}} = 0$$

$$\dot{\boldsymbol{x}}^2 = 2E_x + \frac{g}{z}\boldsymbol{x}^2 \qquad \dot{\boldsymbol{y}}^2 = 2E_y + \frac{g}{z}\boldsymbol{y}^2$$

$$\frac{g}{2zE_x}\boldsymbol{x}^2 + \frac{g}{2zE_y}\boldsymbol{y}^2 + 1 = 0$$

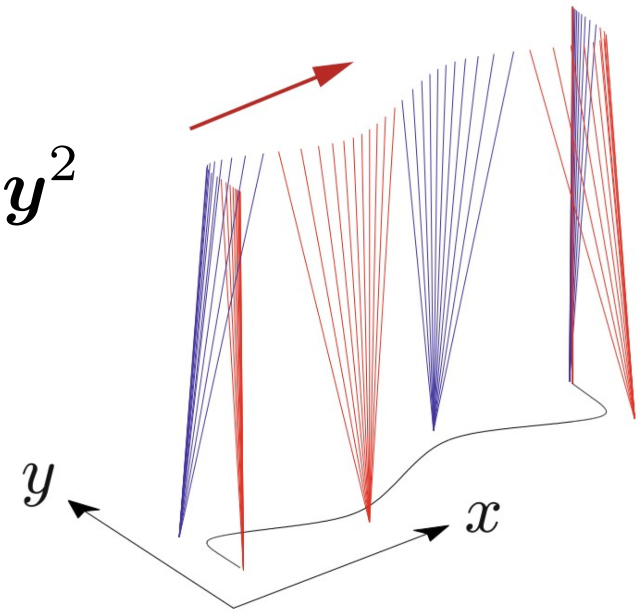
Linear Inverted Pendulum Model 3D

$$\frac{\partial E'_x}{\partial \theta} = -A\cos 2\theta + \frac{B}{2}\sin 2\theta \qquad \theta = 0$$

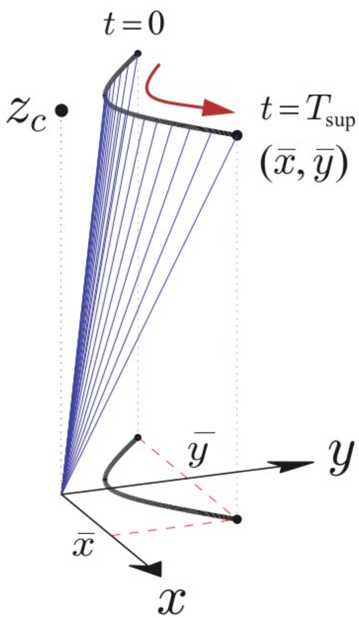
$$\frac{g}{z}\boldsymbol{x}\boldsymbol{y} - \dot{\boldsymbol{x}}\dot{\boldsymbol{y}} = 0$$

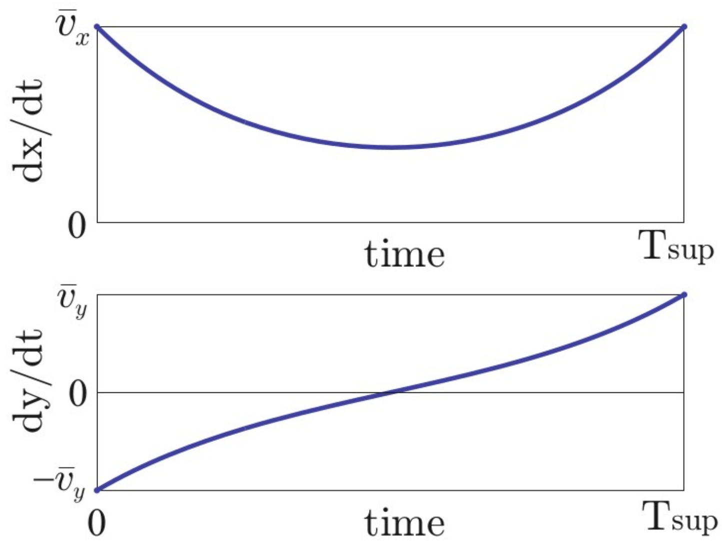
$$\dot{\boldsymbol{x}}^2 = 2E_x + \frac{g}{z}\boldsymbol{x}^2 \qquad \dot{\boldsymbol{y}}^2 = 2E_y + \frac{g}{z}\boldsymbol{y}^2$$

$$\frac{g}{2zE_x}\boldsymbol{x}^2 + \frac{g}{2zE_y}\boldsymbol{y}^2 + 1 = 0$$



Linear Inverted Pendulum Model 3D





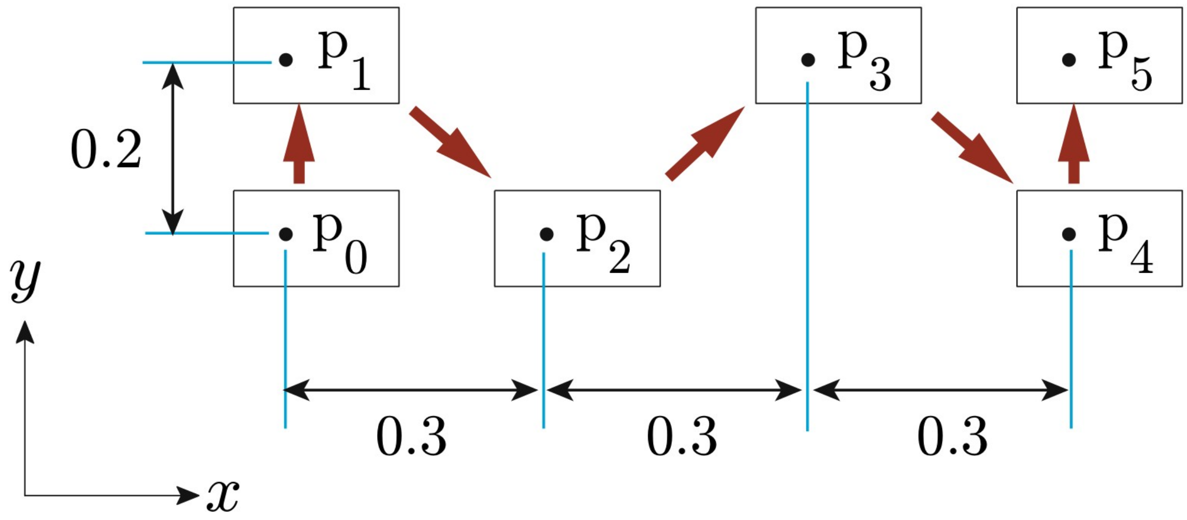
$$\overline{\boldsymbol{x}} = -\overline{\boldsymbol{x}} \cosh\left(\frac{T_{sup}}{T_c}\right) + T_c \overline{\boldsymbol{v}}_x \sinh\left(\frac{T_{sup}}{T_c}\right)$$

$$\overline{\boldsymbol{v}}_x = \frac{\overline{\boldsymbol{x}}\left(\cosh\frac{T_{sup}}{T_c} + 1\right)}{T_c \sinh\frac{T_{sup}}{T_c}}$$

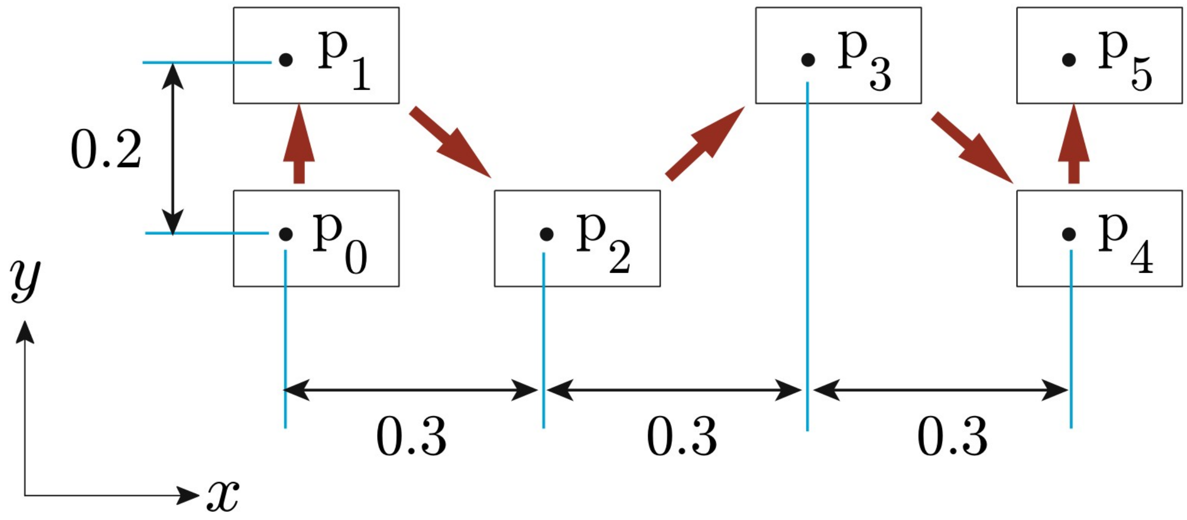
$$\overline{\boldsymbol{y}} = \overline{\boldsymbol{y}} \cosh\left(\frac{T_{sup}}{T_c}\right) - T_c \overline{\boldsymbol{v}}_y \sinh\left(\frac{T_{sup}}{T_c}\right)$$

$$\overline{\boldsymbol{v}}_y = \frac{\overline{\boldsymbol{y}}\left(\cosh\frac{T_{sup}}{T_c} - 1\right)}{T_c \sinh\frac{T_{sup}}{T_c}}$$

Walking Primitives Generation

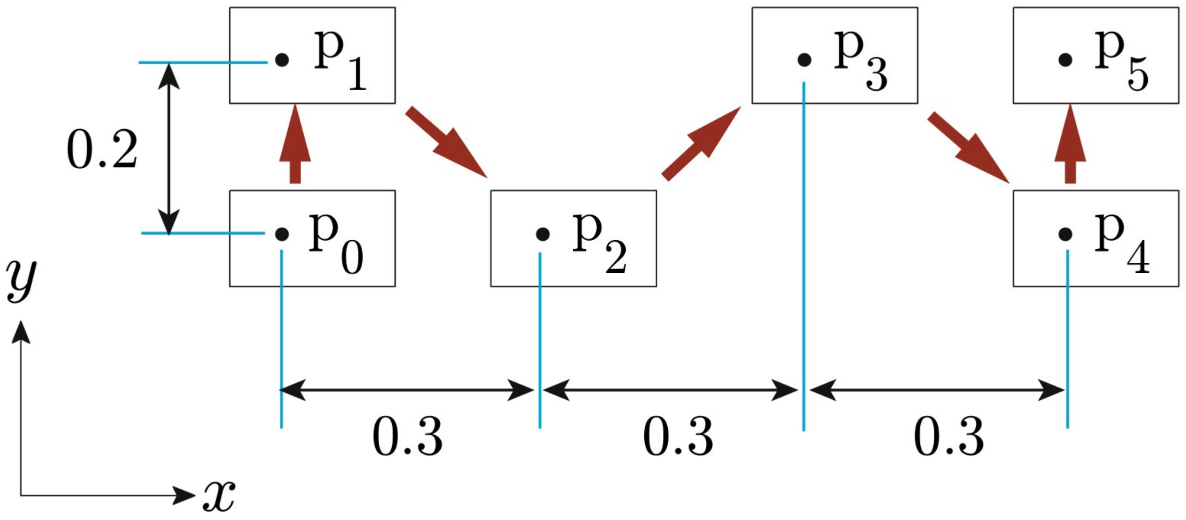


Walking Primitives Generation



$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} + s_x^{(n)} \\ p_y^{(n-1)} - (-1)^n s_y^{(n)} \end{bmatrix}$$

Walking Primitives Generation



$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} + s_x^{(n)} \\ p_y^{(n-1)} - (-1)^n s_y^{(n)} \end{bmatrix}$$

n	1	2	3	4	5
$s_x^{(n)}$	0.0	0.3	0.3	0.3	0
$s_y^{(n)}$	0.2	0.2	0.2	0.2	0.2

Walking Primitives Generation

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix}$$

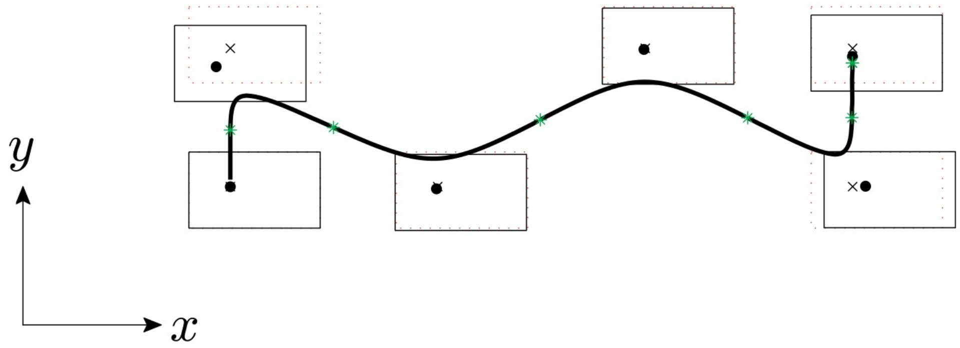
$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} (C+1)/(T_c S)\bar{x}^{(n)} \\ (C-1)/(T_c S)\bar{y}^{(n)} \end{bmatrix} \qquad C \equiv \cosh\frac{T_{sup}}{T_c} \qquad S \equiv \sinh\frac{T_{sup}}{T_c}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Walking Primitives Generation

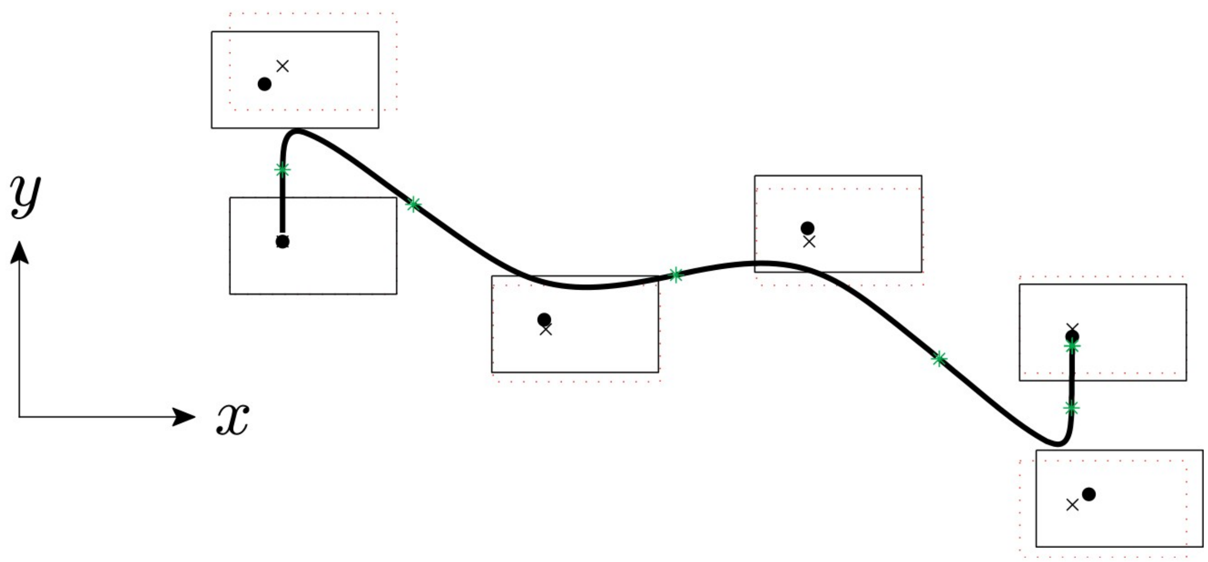
$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix}$$

$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} (C+1)/(T_c S)\bar{x}^{(n)} \\ (C-1)/(T_c S)\bar{y}^{(n)} \end{bmatrix} \qquad C \equiv \cosh \frac{T_{sup}}{T_c} \qquad S \equiv \sinh \frac{T_{sup}}{T_c}$$

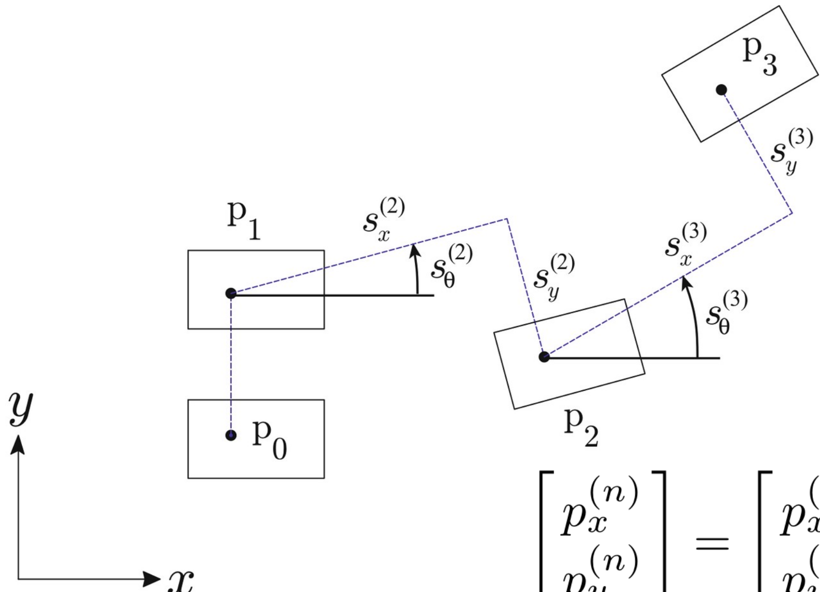


Walking Primitives Generation

n	1	2	3	4	5
$s_x^{(n)}$	0.0	0.2	0.2	0.2	0
$s_y^{(n)}$	0.2	0.3	0.1	0.3	0.2



Walking Primitives Generation



n	1	2	3	4	5
s_x	0.0	0.25	0.25	0.25	0
s_y	0.2	0.2	0.2	0.2	0.2
s_θ	0	20	40	60	60

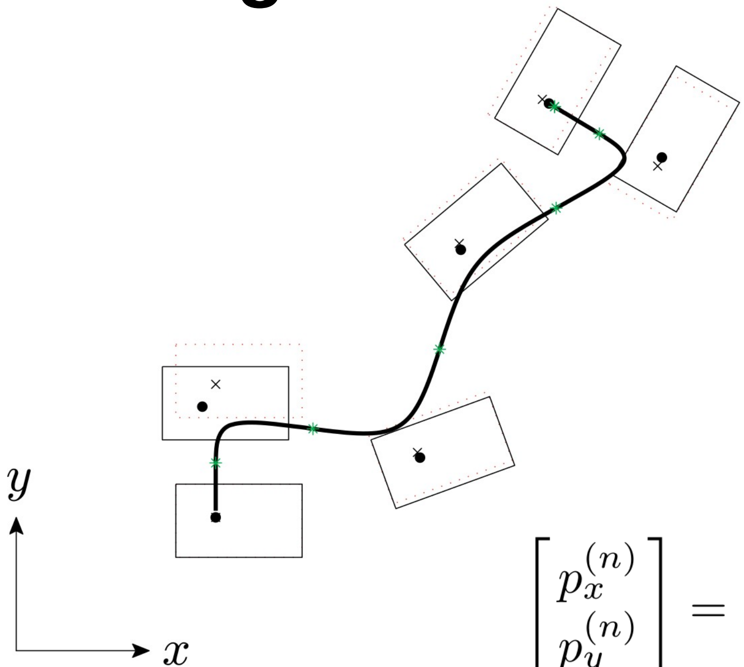
$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} \\ p_y^{(n-1)} \end{bmatrix} + \begin{bmatrix} \cos s_\theta^{(n)} & -\sin s_\theta^{(n)} \\ \sin s_\theta^{(n)} & \cos s_\theta^{(n)} \end{bmatrix} \begin{bmatrix} s_x^{(n)} \\ -(-1)^n s_y^{(n)} \end{bmatrix}$$

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix}$$

$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} (1+C)/(T_c S)\bar{x}^{(n)} \\ (C-1)/(T_c S)\bar{y}^{(n)} \end{bmatrix}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Walking Primitives Generation



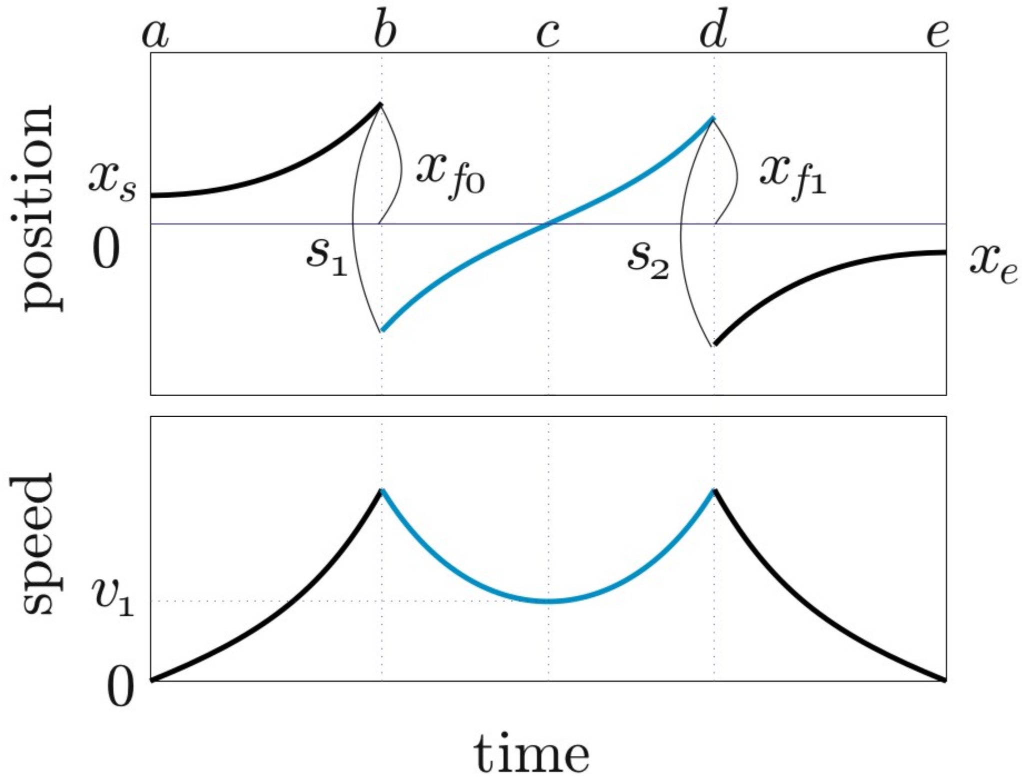
n	1	2	3	4	5
s_x	0.0	0.25	0.25	0.25	0
s_y	0.2	0.2	0.2	0.2	0.2
s_θ	0	20	40	60	60

$$\begin{bmatrix} p_x^{(n)} \\ p_y^{(n)} \end{bmatrix} = \begin{bmatrix} p_x^{(n-1)} \\ p_y^{(n-1)} \end{bmatrix} + \begin{bmatrix} \cos s_\theta^{(n)} & -\sin s_\theta^{(n)} \\ \sin s_\theta^{(n)} & \cos s_\theta^{(n)} \end{bmatrix} \begin{bmatrix} s_x^{(n)} \\ -(-1)^n s_y^{(n)} \end{bmatrix}$$

$$\begin{bmatrix} \bar{x}^{(n)} \\ \bar{y}^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} s_x^{(n+1)}/2 \\ (-1)^n s_y^{(n+1)}/2 \end{bmatrix}$$

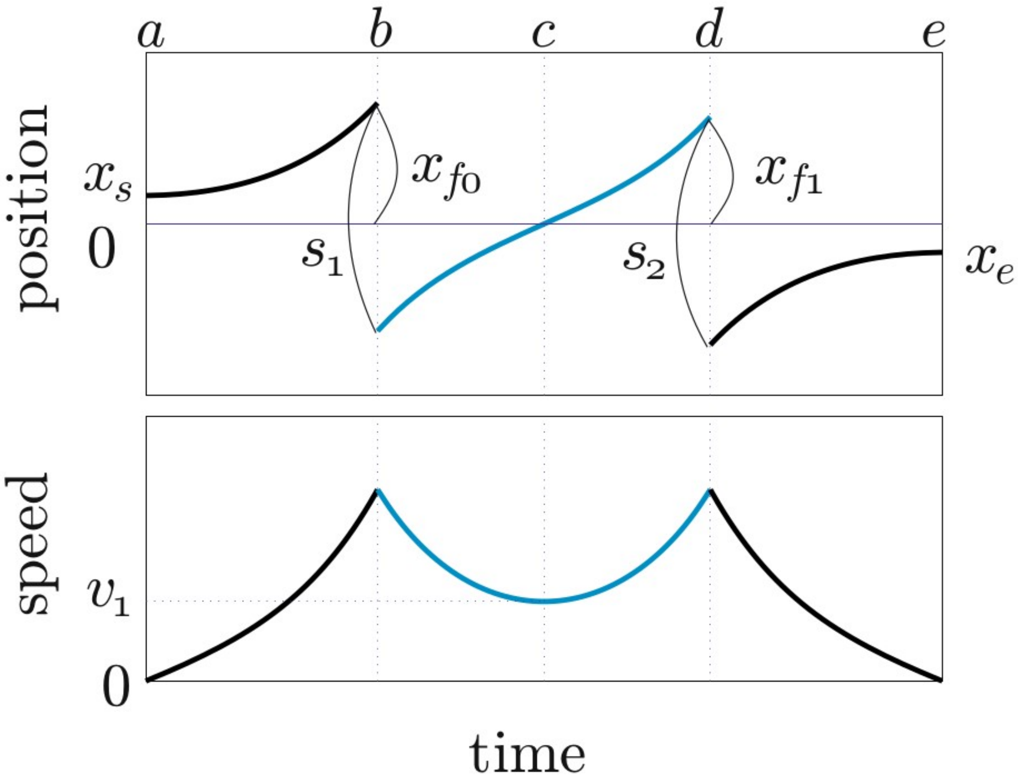
$$\begin{bmatrix} \bar{v}_x^{(n)} \\ \bar{v}_y^{(n)} \end{bmatrix} = \begin{bmatrix} \cos s_\theta^{(n+1)} & -\sin s_\theta^{(n+1)} \\ \sin s_\theta^{(n+1)} & \cos s_\theta^{(n+1)} \end{bmatrix} \begin{bmatrix} (1+C)/(T_c S)\bar{x}^{(n)} \\ (C-1)/(T_c S)\bar{y}^{(n)} \end{bmatrix}$$

Walking Primitives Generation



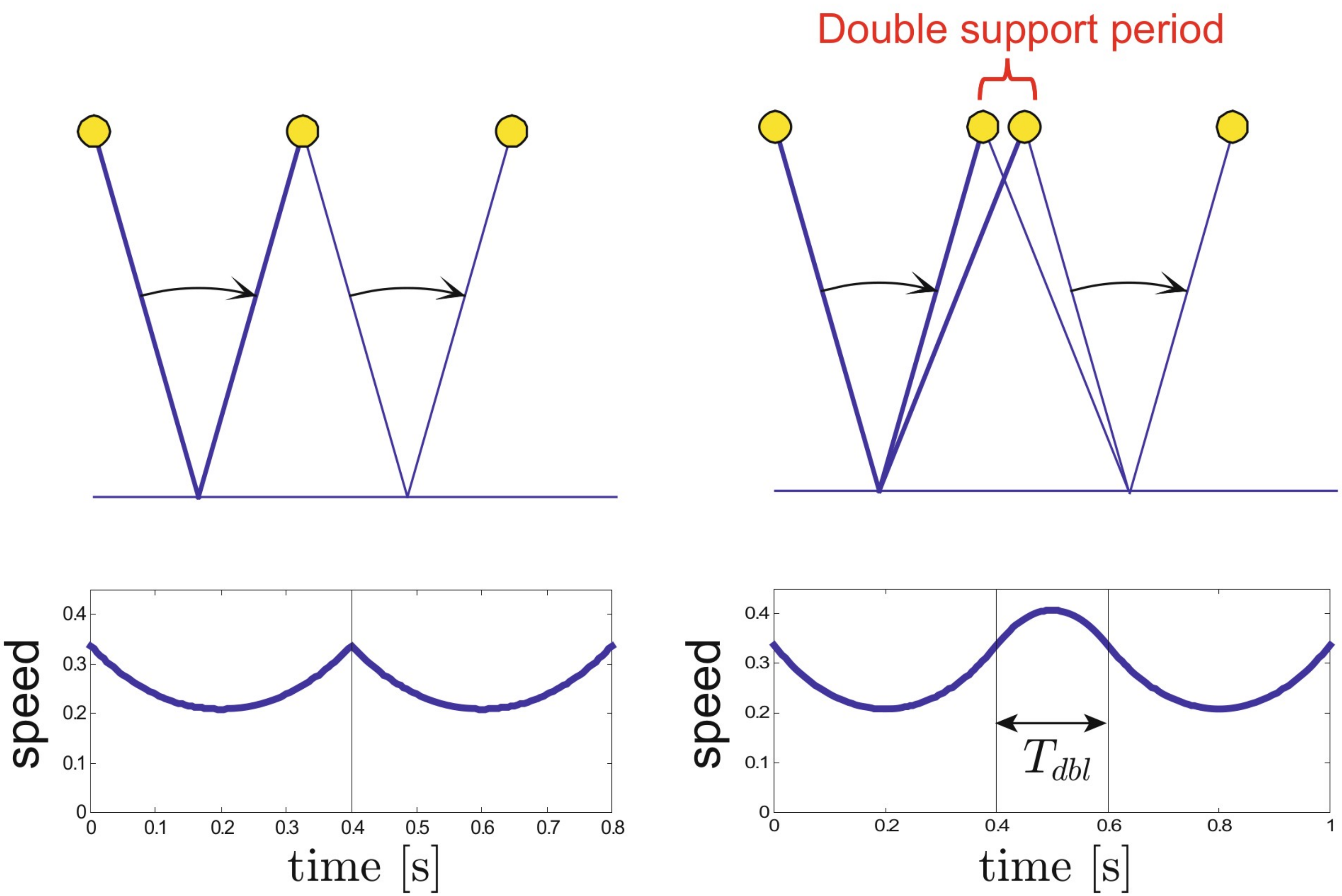
Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Walking Primitives Generation



This is not continuous !!!

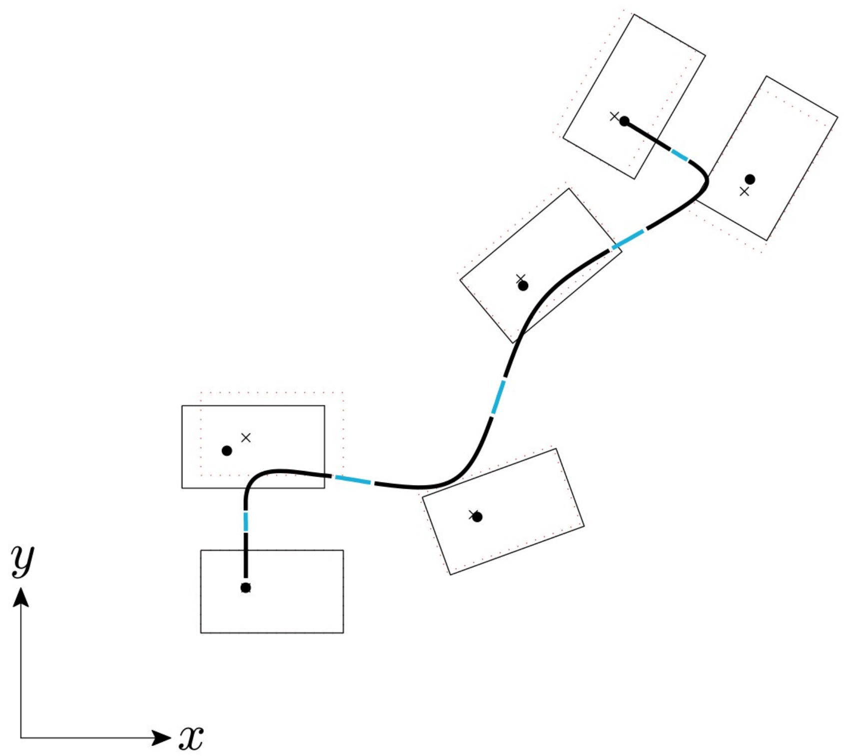
Walking Primitives Generation



Walking Primitives Generation

4th order polynomial (spline) to set initial and final position and velocity.

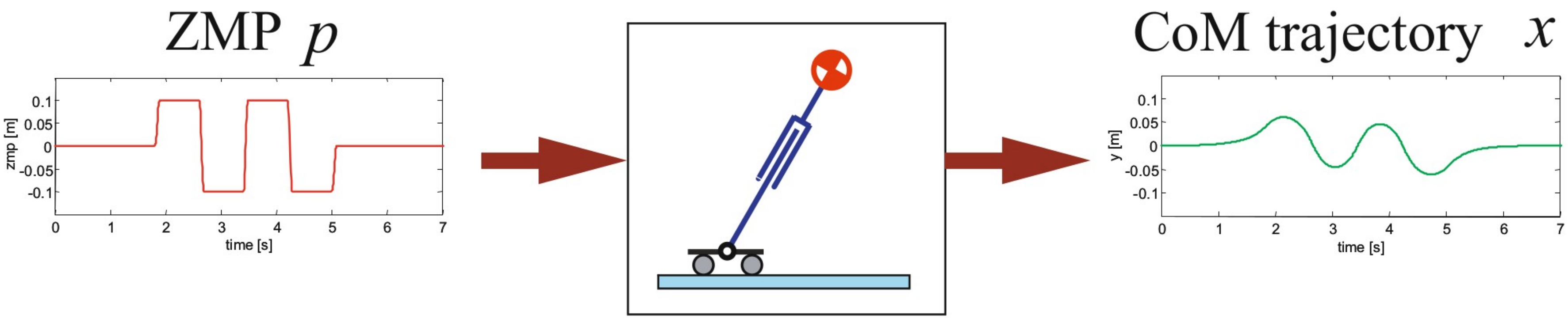
$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4$$



Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

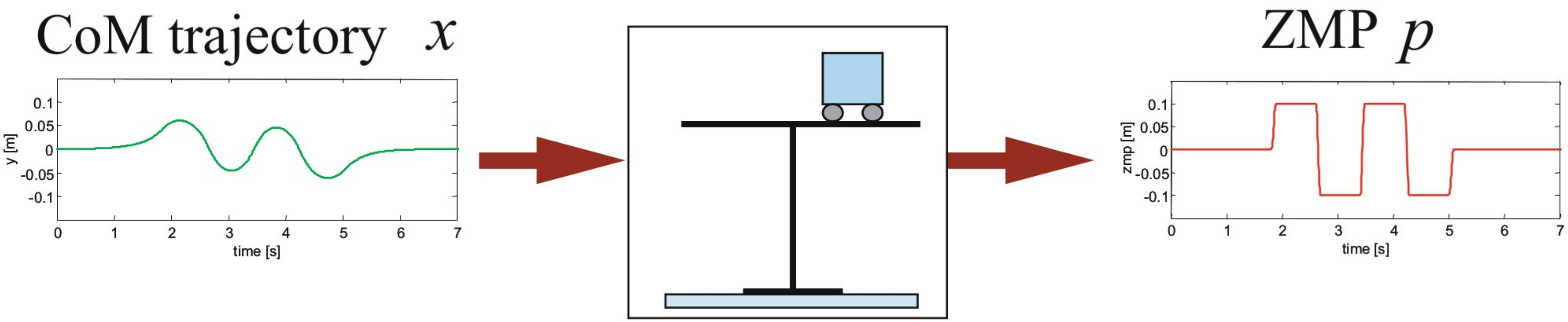
Walking Primitives Generation

- Footsteps used to generate primitives.

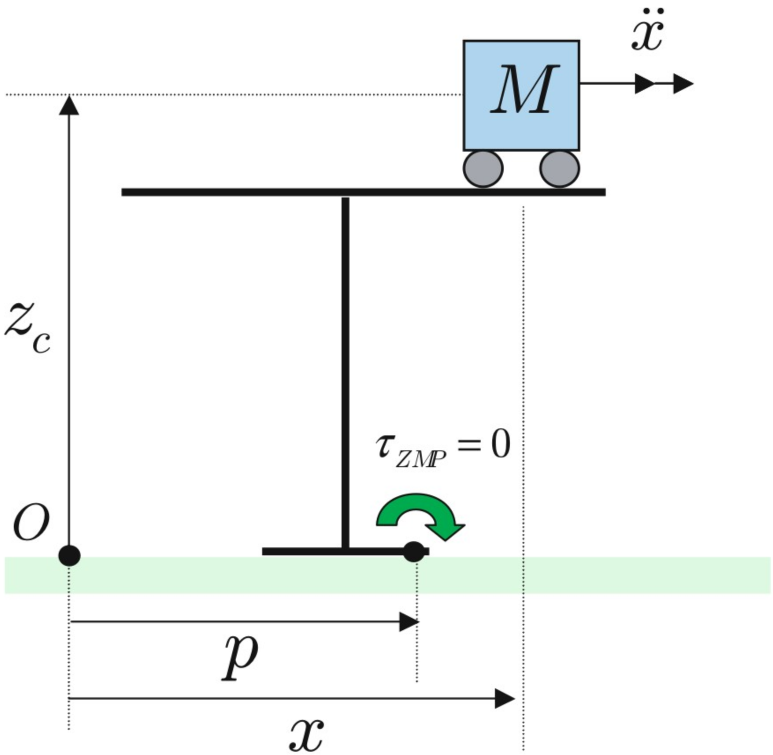


Walking Primitives Generation

- CoM motions to generate ZMP trajectory.

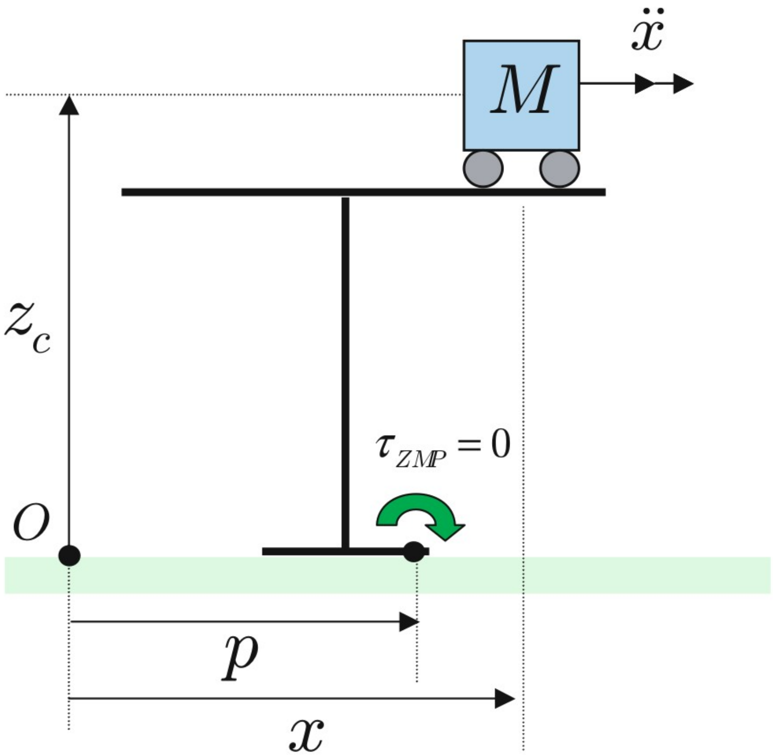


ZMP Based Walking Motion



$$\boldsymbol{p} = \boldsymbol{x} - \frac{z}{g}\ddot{\boldsymbol{x}} \qquad \ddot{\boldsymbol{x}} = \frac{g}{z}(\boldsymbol{x} - \boldsymbol{p})$$

ZMP Based Walking Motion



$$\boldsymbol{p} = \boldsymbol{x} - \frac{z}{g}\ddot{\boldsymbol{x}} \qquad \ddot{\boldsymbol{x}} = \frac{g}{z}(\boldsymbol{x} - \boldsymbol{p})$$

$$\ddot{x}_i = \frac{x_{i-1} - 2x_i + x_{i+1}}{\Delta t^2}$$

$$x_i \equiv x(i\Delta t)$$

$$p_i = ax_{i-1} + bx_i + cx_{i+1}$$

$$a_i \equiv -z_c/(g\Delta t^2)$$

$$b_i \equiv 2z_c/(g\Delta t^2) + 1$$

$$c_i \equiv -z_c/(g\Delta t^2)$$

ZMP Based Walking Motion

$$
\begin{bmatrix} p_1' \\ p_2 \\ \vdots \\ p_{N-1} \\ p_N' \end{bmatrix} = \begin{bmatrix} a_1+b_1 & c_1 & 0 & & & \\ a_2 & b_2 & c_2 & \ddots & & \\ & & & \ddots & & \\ & & & \ddots & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & & 0 & a_N & b_N+c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}
$$

$$
p_1' = p_1 + a_1 v_1 \Delta t
$$

$$
p_N' = p_N - c_N v_N \Delta t
$$

ZMP Based Walking Motion

$$\begin{bmatrix} p'_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p'_N \end{bmatrix} = \begin{bmatrix} a_1 + b_1 & c_1 & 0 & & & \\ a_2 & b_2 & c_2 & \ddots & & \\ & & & \ddots & & \\ & & & \ddots & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & & 0 & a_N & b_N + c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}$$

$$p'_1 = p_1 + a_1 v_1 \Delta t$$

$$p'_N = p_N - c_N v_N \Delta t$$

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{x}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

ZMP Based Walking Motion

$$
\begin{bmatrix} p'_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p'_N \end{bmatrix} = \begin{bmatrix} a_1 + b_1 & c_1 & 0 & & & \\ a_2 & b_2 & c_2 & \ddots & & \\ & & & \ddots & & \\ & & & \ddots & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & & 0 & a_N & b_N + c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}
$$

$$p'_1 = p_1 + a_1 v_1 \Delta t$$

$$p'_N = p_N - c_N v_N \Delta t$$

$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{x}$$

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{p}$$

ZMP Based Walking Motion

$$
\begin{bmatrix} p'_1 \\ p_2 \\ \vdots \\ p_{N-1} \\ p'_N \end{bmatrix} = \begin{bmatrix} a_1 + b_1 & c_1 & 0 & & & \\ a_2 & b_2 & c_2 & \ddots & & \\ & & & \ddots & & \\ & & & \ddots & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & & 0 & a_N & b_N + c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}
$$

$$p'_1 = p_1 + a_1 v_1 \Delta t$$

$$p'_N = p_N - c_N v_N \Delta t$$

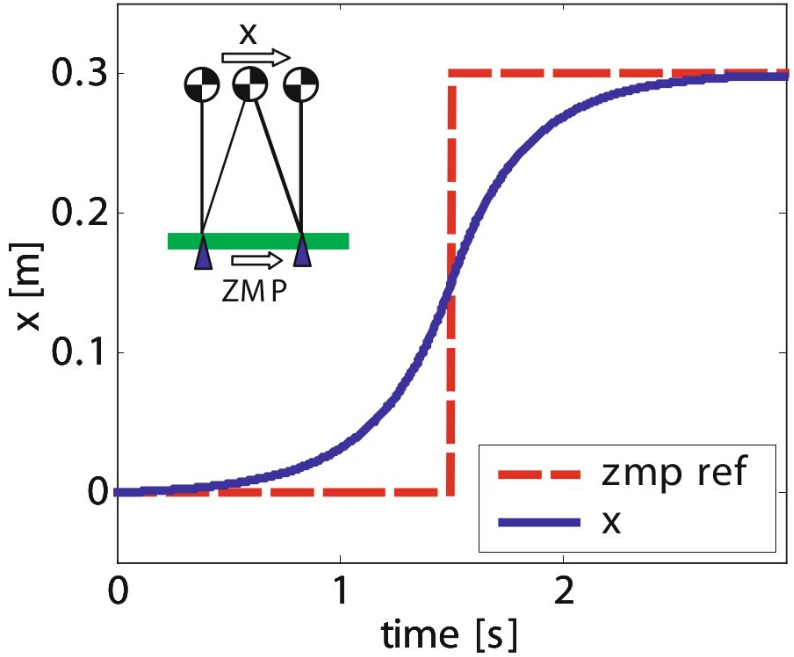
$$\boldsymbol{p} = \boldsymbol{A}\boldsymbol{x}$$

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{p}$$

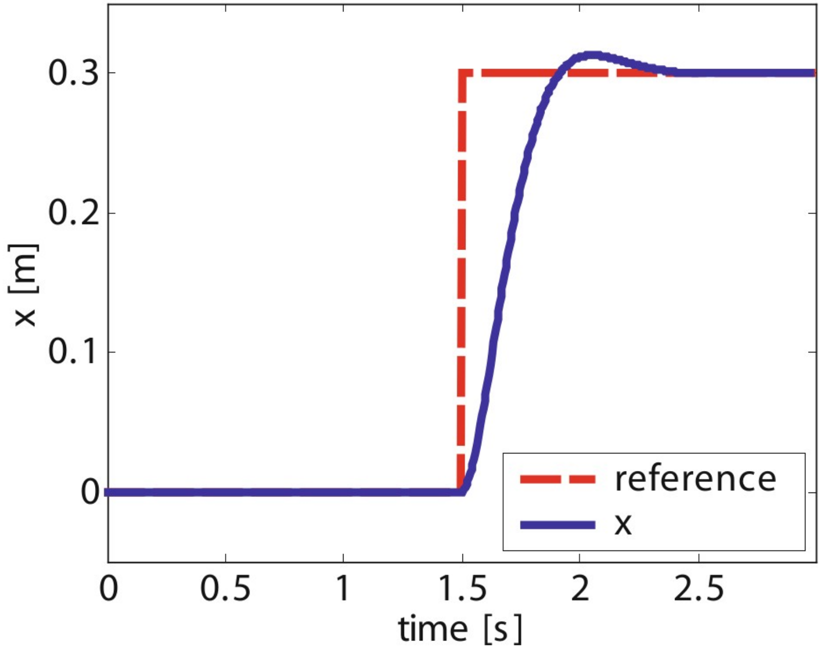
$$\Delta\boldsymbol{x} = \boldsymbol{A}^{-1}(\boldsymbol{p}^* - \boldsymbol{p}^d)$$

$$\boldsymbol{x} := \boldsymbol{x} - \Delta\boldsymbol{x}$$

Model Predictive Control



Walking motion



Classic Servoing

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Model Predictive Control

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{b}u_k \\ p_k = \boldsymbol{c}\boldsymbol{x}_k \end{cases}$$

$$\boldsymbol{x}_k \equiv [\, x(k\Delta t) \; \dot{x}(k\Delta t) \; \ddot{x}(k\Delta t) \,]^T$$

$$u_k \equiv u(k\Delta t),$$

$$p_k \equiv p(k\Delta t),$$

$$\boldsymbol{A} \equiv \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{b} \equiv \begin{bmatrix} \Delta t^3/6 \\ \Delta t^2/2 \\ \Delta t \end{bmatrix}$$

$$\boldsymbol{c} \equiv [\, 1 \; 0 \; -z_c/g \,]$$

Minimize the performance index

$$J = \sum_{j=1}^{\infty} \{ Q(p_j^{ref} - p_j)^2 + R u_j^2 \}$$

$$Q, \; R \in \mathbb{R}^+$$ Weights

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

Model Predictive Control

Minimize the tracking control problem

$$J = \sum_{j=1}^{\infty} \{Q(p_j^{ref} - p_j)^2 + Ru_j^2\}$$

$$Q,\ R \in \mathbb{R}^+$$ Weights

$$u_k = -\boldsymbol{K}x_k + [f_1, f_2, \cdots f_N] \begin{bmatrix} p_{k+1}^{ref} \\ \vdots \\ p_{k+N}^{ref} \end{bmatrix}$$

$$\boldsymbol{K} \equiv (R + \boldsymbol{b}^T \boldsymbol{P} \boldsymbol{b})^{-1} \boldsymbol{b}^T \boldsymbol{P} \boldsymbol{A}$$
$$f_i \equiv (R + \boldsymbol{b}^T \boldsymbol{P} \boldsymbol{b})^{-1} \boldsymbol{b}^T (\boldsymbol{A} - \boldsymbol{b}\boldsymbol{K})^{T*(i-1)} \boldsymbol{c}^T Q$$

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P} \boldsymbol{A} + \boldsymbol{c}^T Q \boldsymbol{c} - \boldsymbol{A}^T \boldsymbol{P} \boldsymbol{b} (R + \boldsymbol{b}^T \boldsymbol{P} \boldsymbol{b})^{-1} \boldsymbol{b}^T \boldsymbol{P} \boldsymbol{A}$$ → Riccati Equation

Model Predictive Control

Minimize the tracking control problem

$$J = \sum_{j=1}^{\infty} \{Q(p_j^{ref} - p_j)^2 + Ru_j^2\}$$

$$Q,\ R \in \mathbb{R}^+$$ Weights

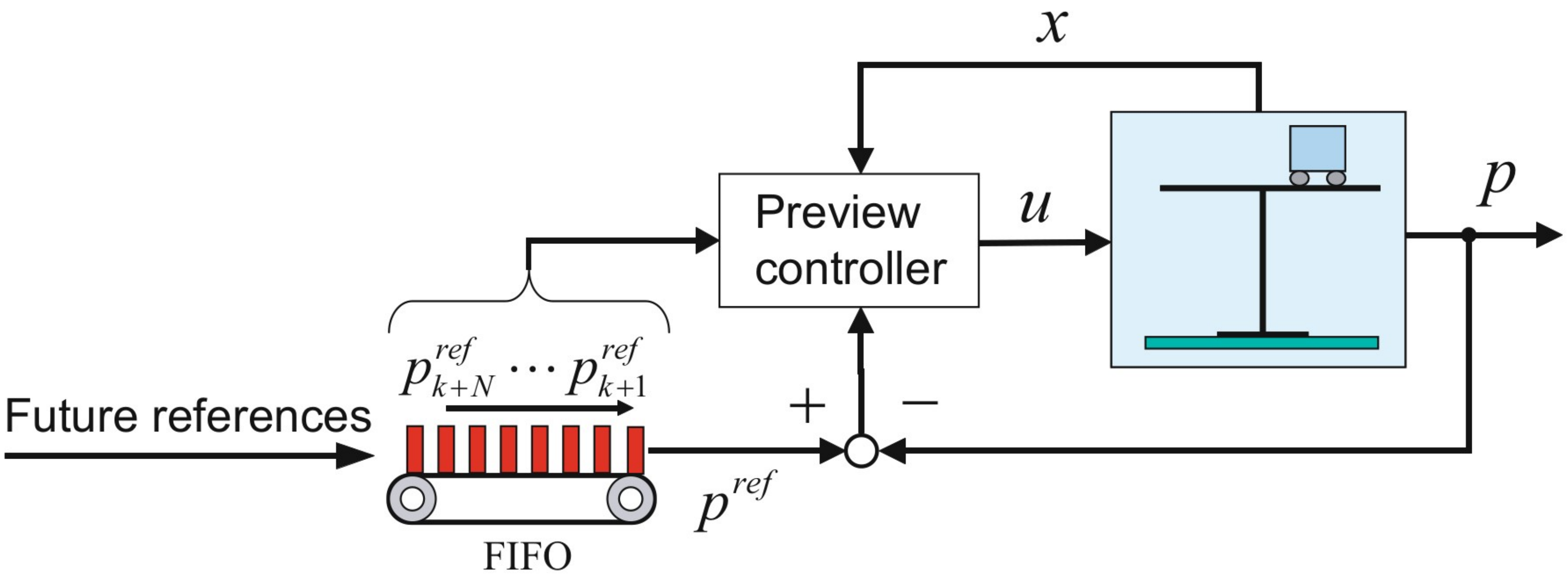
$$u_k = -\boldsymbol{K}x_k + [f_1, f_2, \cdots f_N] \begin{bmatrix} p_{k+1}^{ref} \\ \vdots \\ p_{k+N}^{ref} \end{bmatrix}$$

LQR Design Problem Solution will produce **P** and **K** e. g. dlqr in Matlab

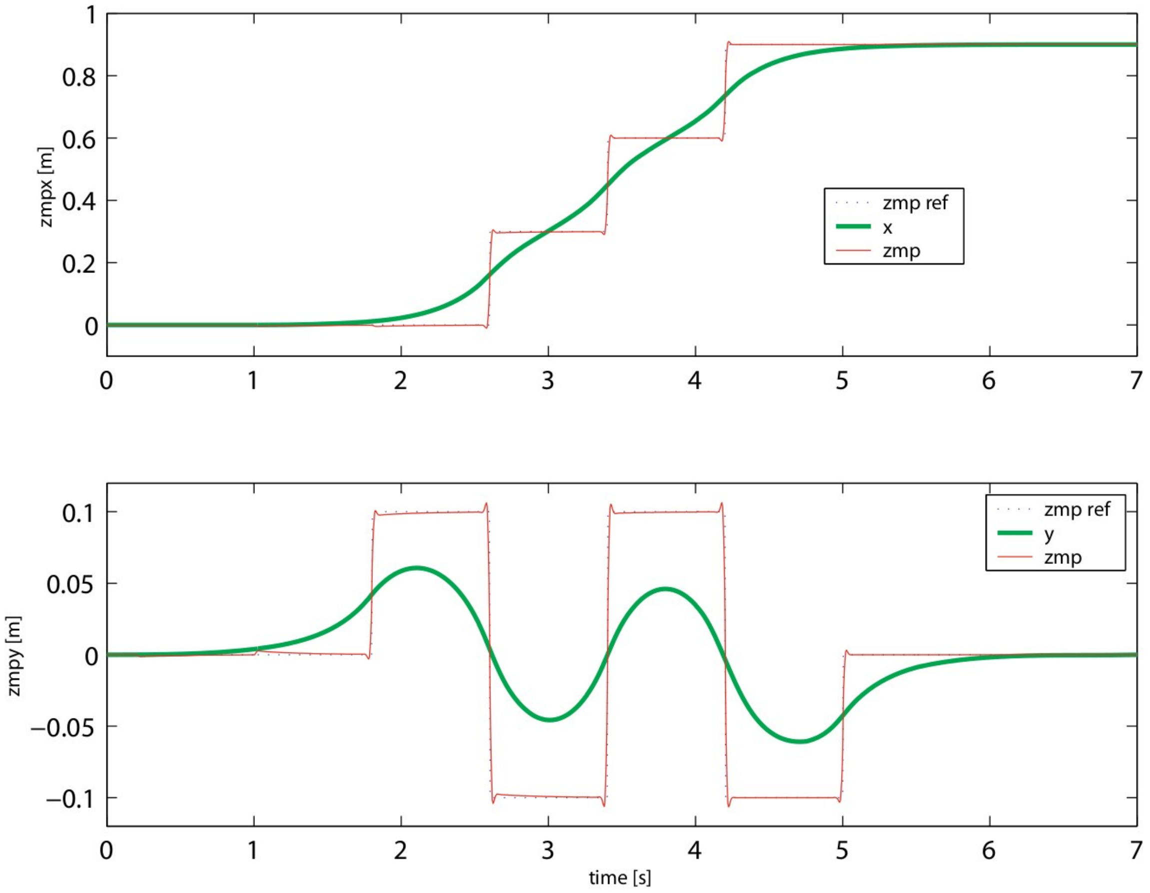
$$\boldsymbol{K} \equiv (R + \boldsymbol{b}^T\boldsymbol{P}\boldsymbol{b})^{-1}\boldsymbol{b}^T\boldsymbol{P}\boldsymbol{A}$$
$$f_i \equiv (R + \boldsymbol{b}^T\boldsymbol{P}\boldsymbol{b})^{-1}\boldsymbol{b}^T(\boldsymbol{A} - \boldsymbol{b}\boldsymbol{K})^{T*(i-1)}\boldsymbol{c}^T Q$$

$$\boldsymbol{P} = \boldsymbol{A}^T\boldsymbol{P}\boldsymbol{A} + \boldsymbol{c}^T Q\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{P}\boldsymbol{b}(R + \boldsymbol{b}^T\boldsymbol{P}\boldsymbol{b})^{-1}\boldsymbol{b}^T\boldsymbol{P}\boldsymbol{A}$$ → Riccati Equation

Model Predictive Control

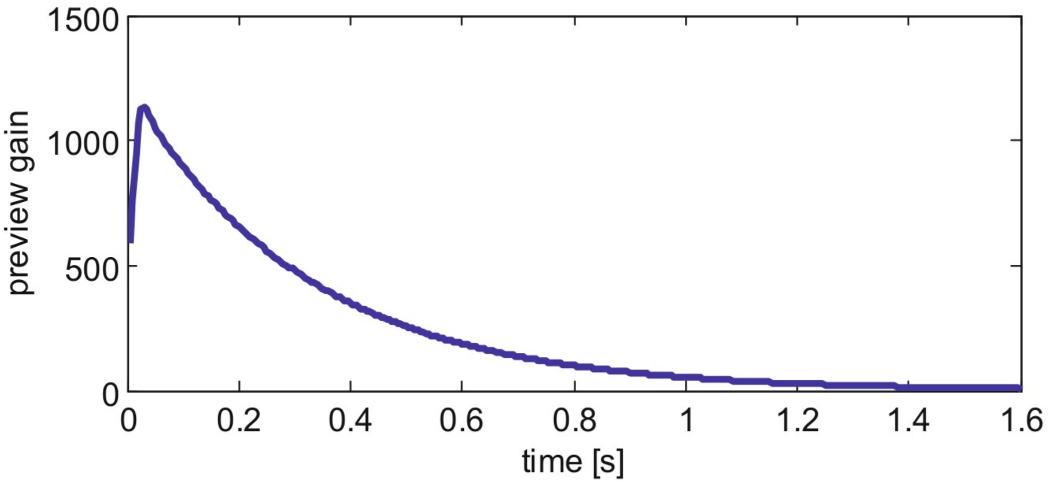


Model Predictive Control



Model Predictive Control

Preview gain



$$\sum_{i=j}^{N+j} \tilde{f}_i$$

DCM Based Walking Motions Generation

$$\boldsymbol{\xi} = \boldsymbol{x} + \frac{\dot{\boldsymbol{x}}}{\omega}$$

$$\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{x}} + \frac{\ddot{\boldsymbol{x}}}{\omega}$$

$$\dot{\boldsymbol{\xi}} = \omega(\boldsymbol{\xi} - \mathbf{p})$$

$$\dot{\boldsymbol{x}} = -\omega(\boldsymbol{x} - \boldsymbol{\xi})$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

DCM Based Walking Motions Generation

$$\boldsymbol{\xi} = \boldsymbol{x} + \frac{\dot{\boldsymbol{x}}}{\omega}$$

$$\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{x}} + \frac{\ddot{\boldsymbol{x}}}{\omega}$$

$$\dot{\boldsymbol{\xi}} = \omega(\boldsymbol{\xi} - \mathbf{p})$$

$$\dot{\boldsymbol{x}} = -\omega(\boldsymbol{x} - \boldsymbol{\xi})$$

1) Plan footsteps

2) Define Virtual Repellent Points (VRP)

3) Define waypoints for the DCM

4) Interpolate CoM trajectory

DCM Based Walking Motions Generation

$$\boldsymbol{\xi} = \boldsymbol{x} + \frac{\dot{\boldsymbol{x}}}{\omega}$$

$$\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{x}} + \frac{\ddot{\boldsymbol{x}}}{\omega}$$

$$\dot{\boldsymbol{\xi}} = \omega(\boldsymbol{\xi} - \mathbf{p})$$

$$\dot{\boldsymbol{x}} = -\omega(\boldsymbol{x} - \boldsymbol{\xi})$$

1) Plan footsteps

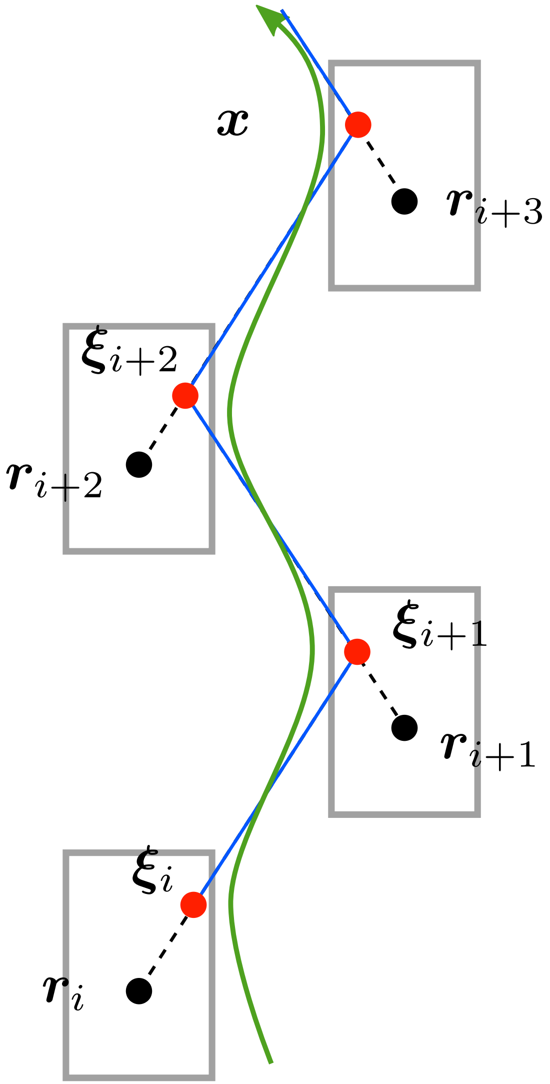
2) Define Virtual Repellent Points (VRP)

3) Define waypoints for the DCM

4) Interpolate CoM trajectory

[Englsberger et al., 2011]

DCM Based Walking Motions Generation



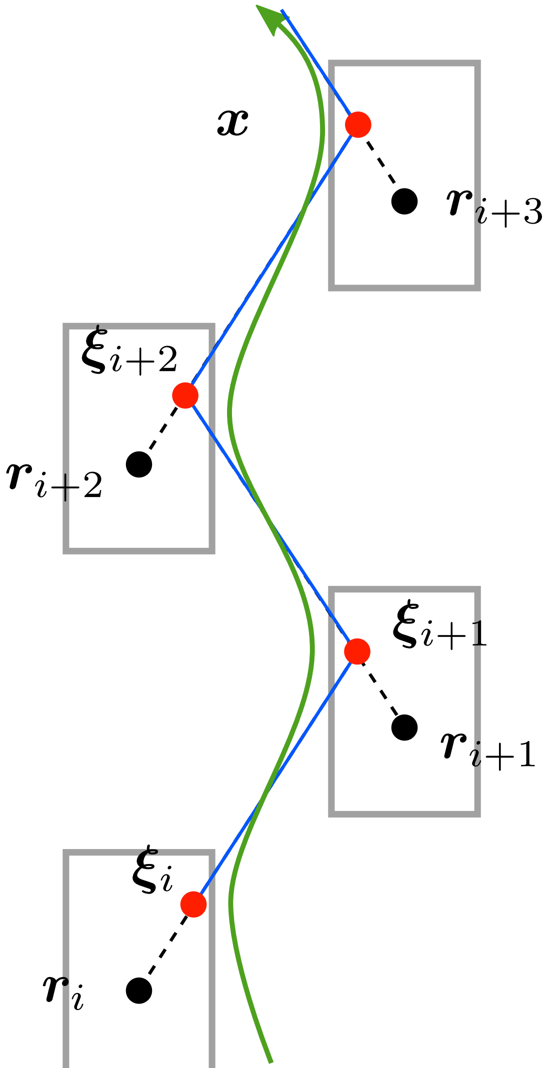
1) Define Footsteps

Recommended at least a horizon of 3 steps.

$$\boldsymbol{r}_1,\ \boldsymbol{r}_2,\ \cdots\ \boldsymbol{r}_N,$$

Including side and angular steps

DCM Based Walking Motions Generation

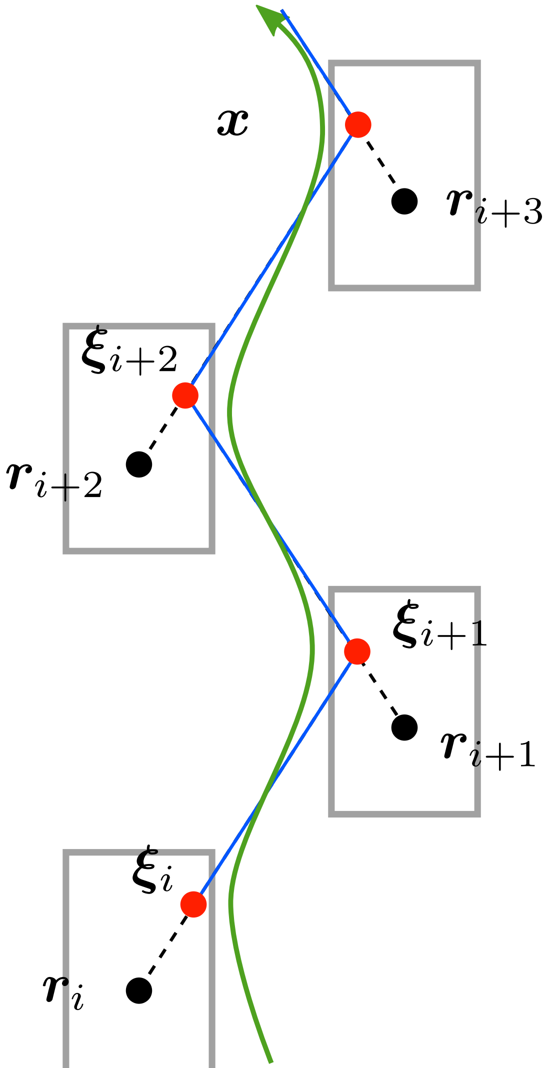


2) Define Virtual Repellent Points (VRP)

$$\boldsymbol{r}_i = {}^{\mathbf{w}}\,\mathbf{r}_i + \begin{bmatrix} 0 \\ 0 \\ z_x \end{bmatrix}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

DCM Based Walking Motions Generation



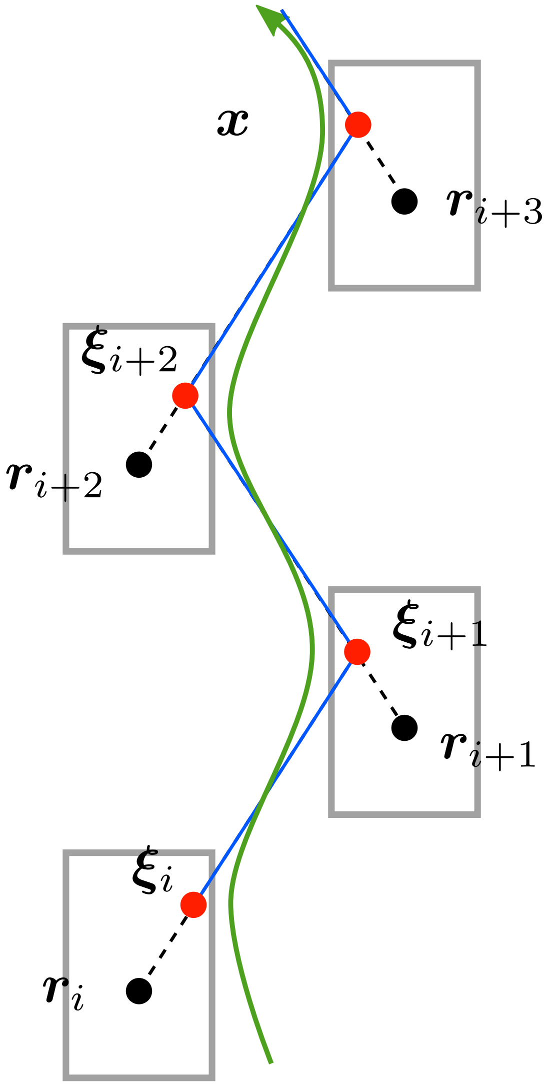
3) Define way-points for the DCM

$$\boldsymbol{\xi}_i = \boldsymbol{r}_{i+1} + e^{\omega t_{step}} \left(\boldsymbol{\xi}_{i+1} + \boldsymbol{r}_{i+1} \right)$$

$$t_{step} = \frac{\omega}{\pi} \qquad \boldsymbol{\omega} = \sqrt{\frac{g}{z}}$$

Chair for Cognitive Systems
School of Computation, Information and Technology
Technische Universität München

DCM Based Walking Motions Generation

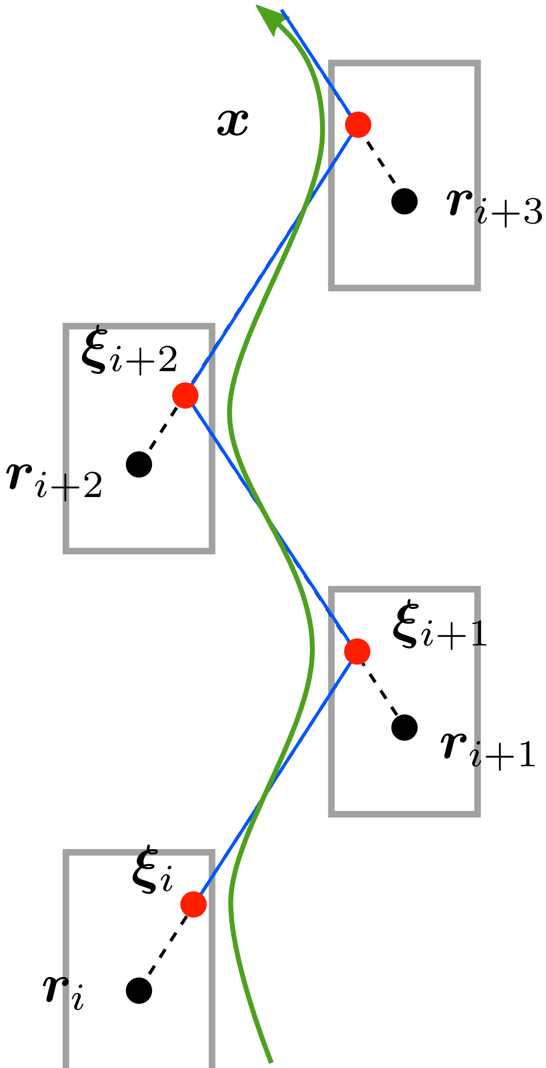


4) Generate CoM Trajectory

$$\boldsymbol{\xi}_r = \boldsymbol{r}_i + e^{\omega(t - t_{step})}\,(\boldsymbol{\xi}_i + \boldsymbol{r}_i)$$

$$\dot{\boldsymbol{x}} = -\omega(\boldsymbol{x} - \boldsymbol{\xi})$$

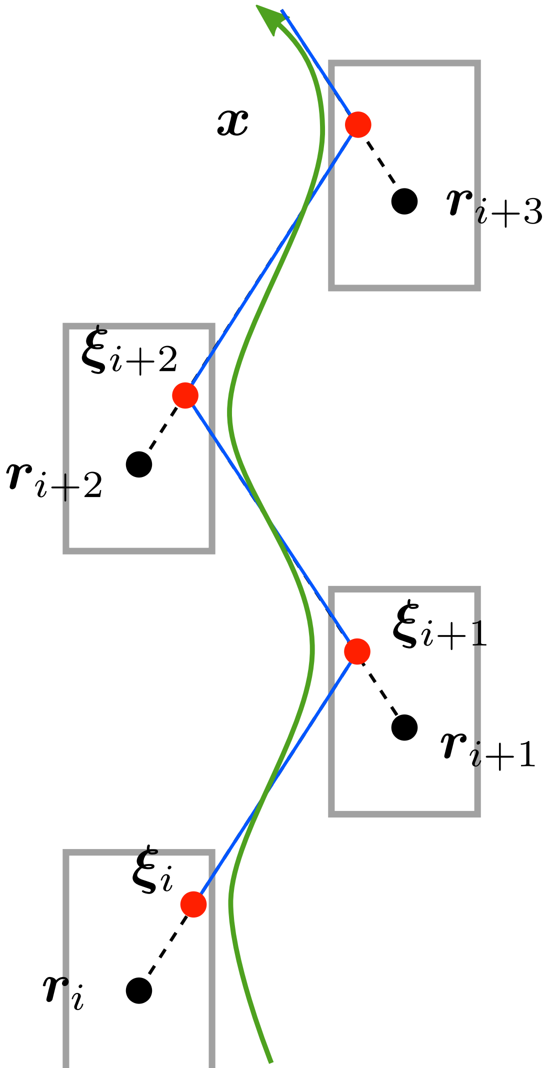
DCM Based Walking Motions Generation



DCM/ZMP balance control

$$\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_r = -k_\xi \left(\boldsymbol{\xi} - \boldsymbol{\xi}_r \right)$$

DCM Based Walking Motions Generation

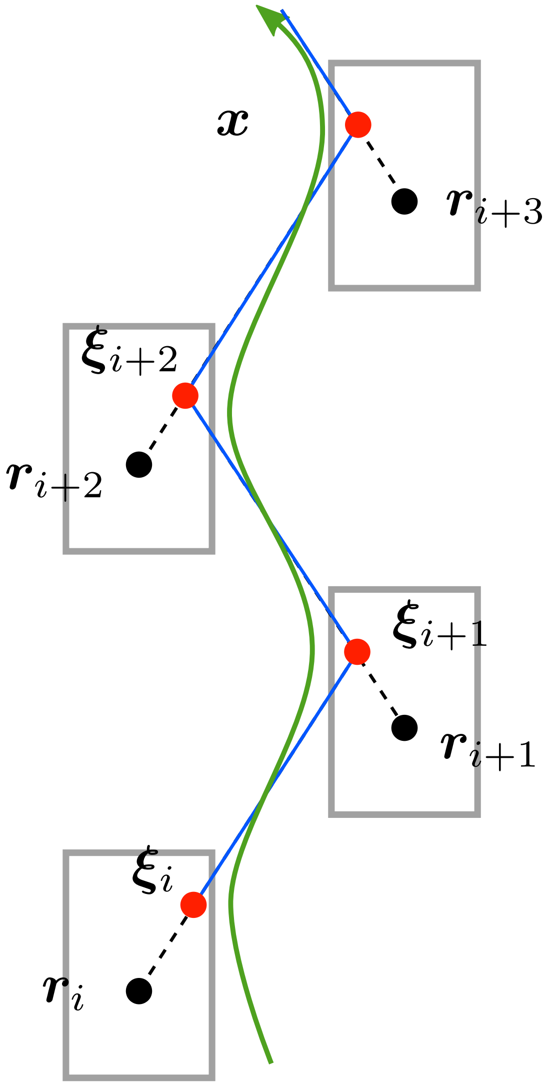


DCM/ZMP balance control

$$\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_r = -k_\xi \left(\boldsymbol{\xi} - \boldsymbol{\xi}_r\right)$$

$$\mathbf{p}_d = \mathbf{p}_r - \left(1 + \frac{k_\xi}{\omega}\right)\left(\boldsymbol{\xi}_r - \boldsymbol{\xi}\right)$$

DCM Based Walking Motions Generation



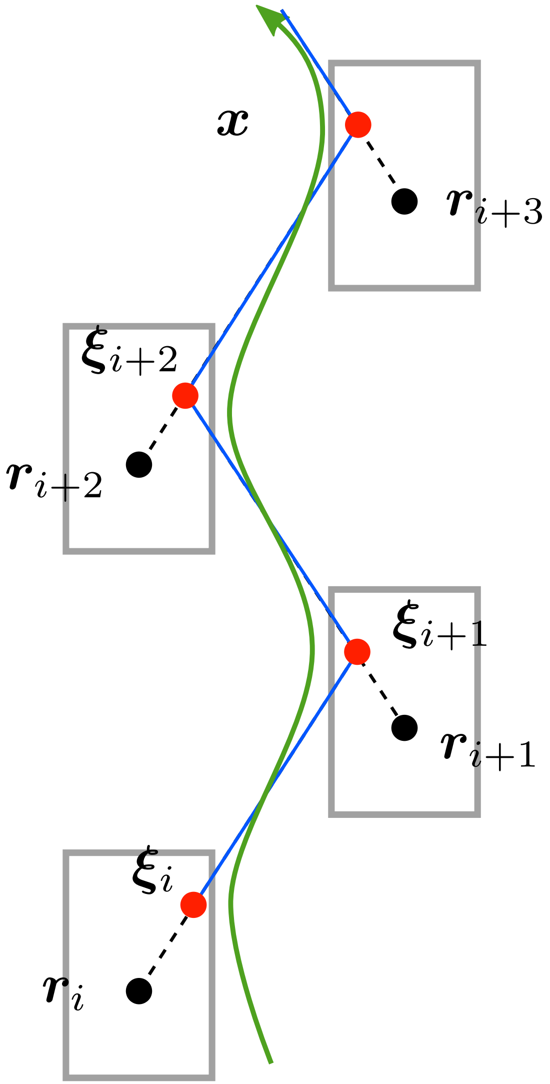
DCM/ZMP balance control

$$\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_r = -k_\xi \left(\boldsymbol{\xi} - \boldsymbol{\xi}_r\right)$$

$$\mathbf{p}_d = \mathbf{p}_r - \left(1 + \frac{k_\xi}{\omega}\right)\left(\boldsymbol{\xi}_r - \boldsymbol{\xi}\right)$$

$$\dot{\mathbf{p}} - \dot{\mathbf{p}}_d = -k_p \left(\mathbf{p} - \mathbf{p}_d\right)$$

DCM Based Walking Motions Generation



DCM/ZMP balance control

$$\dot{\boldsymbol{\xi}} - \dot{\boldsymbol{\xi}}_r = -k_\xi \left(\boldsymbol{\xi} - \boldsymbol{\xi}_r\right)$$

$$\mathbf{p}_d = \mathbf{p}_r - \left(1 + \frac{k_\xi}{\omega}\right)\left(\boldsymbol{\xi}_r - \boldsymbol{\xi}\right)$$

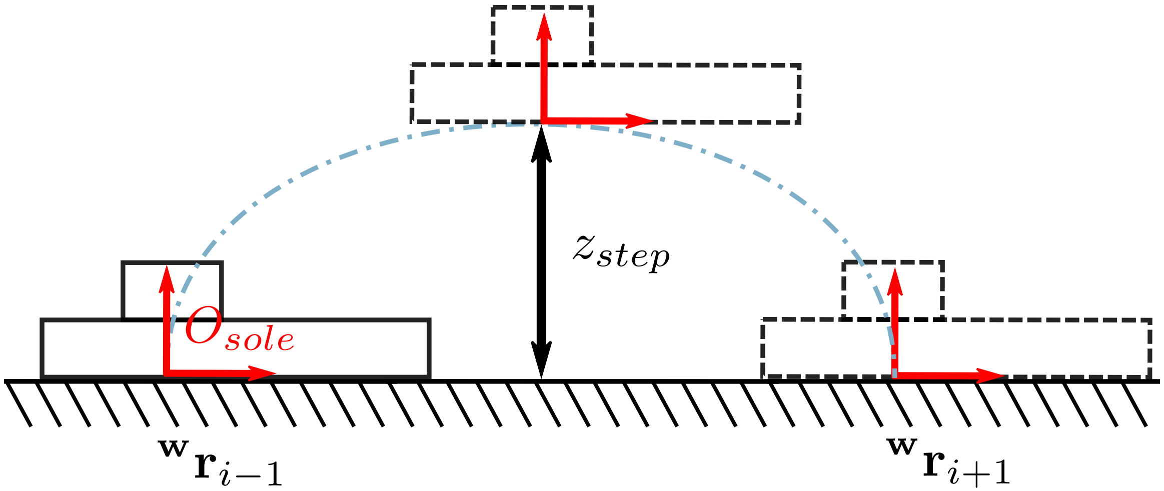
$$\dot{\mathbf{p}} - \dot{\mathbf{p}}_d = -k_p \left(\mathbf{p} - \mathbf{p}_d\right)$$

$$\dot{\boldsymbol{x}}_d = \dot{\mathbf{p}} + k_p(\mathbf{p} - \mathbf{p}_d)$$

Foot Trajectory Generation

$$t_{step} = t_{ss} + t_{ds}$$

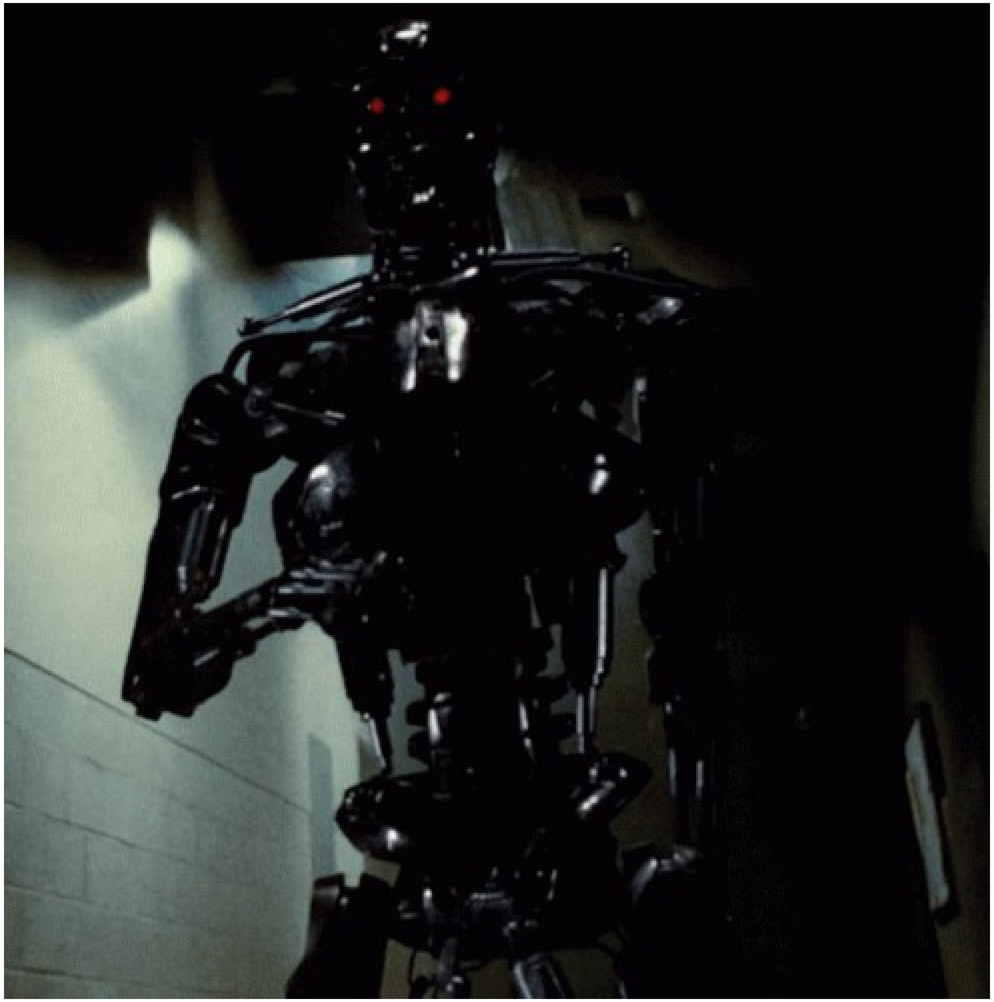
$$\boldsymbol{x}_{foot}(0) = \boldsymbol{r}_{i-1}$$
$$\dot{\boldsymbol{x}}_{foot}(0) = 0$$
$$\ddot{\boldsymbol{x}}_{foot}(0) = 0$$



$$\boldsymbol{x}_{foot}(t_{ss}) = \boldsymbol{r}_{i+1}$$
$$\dot{\boldsymbol{x}}_{foot}(t_{ss}) = 0$$
$$\ddot{\boldsymbol{x}}_{foot}(t_{ss}) = 0$$

$$\boldsymbol{x}\left(\frac{t_{ss}}{2}\right) = \boldsymbol{r_{i-1}} + \frac{1}{2}\left(\boldsymbol{r_{i-1}} + \boldsymbol{r_{i+1}}\right) + \begin{bmatrix} 0 \\ 0 \\ z_{step} \end{bmatrix}$$

Questions?



Next session:

Tutorial 7: Foot Step Planning