HUMANOIDS2022 Tutorial

How to Make Your Robot Balance and Walk in Simulation

Yuichi Tazaki
Kobe University

A motivating question …

Can we make a robot balance and walk robustly by using **simple mechanisms** only ?

without
- optimization
- prediction
- state estimation
- environmental perception

If so, to what extent?

In other words, where is the limit of such approach?

In what circumstances do we need more advanced methods (e.g., MPC)?

Background

Vnoid

a small light-weight library that is used in sample codes of HVAC

https://github.com/ytazz/vnoid

Vnoid is

Not intended to be used as a black-box software library, but as a collection of code that helps to understand basic functionalities of legged robots.

Each function is implemented in about 100 lines of C code.

Easy to read, modify, and copy-and-paste to your own code.

Comes with examples that run on Choreonoid.

It currently provides:

- Forward and inverse kinematics
- Walking pattern generation
- Balance control

Kinematics

Difficulties

- Kinematic singularity
- Joint range limit

Numerical (differential) vs Analytical methods

	Pros	Cons
Numerical	Flexible (kinematic closed-loop, various constraints, prioritization)	Computationally expensive Not so easy to stabilize near singularity
Analytical	Fast, easy to implement Easy to handle singularities	Specialized to each kinematic structure

IK (Inverse Kinematics) specific to Legged Robots

- Center-of-Mass Inverse Kinematics (Kinetics)
- Angular Momentum (Differential) Inverse Kinematics (Kinetics)

Kinematics

Introductory material on kinematics of legged robots

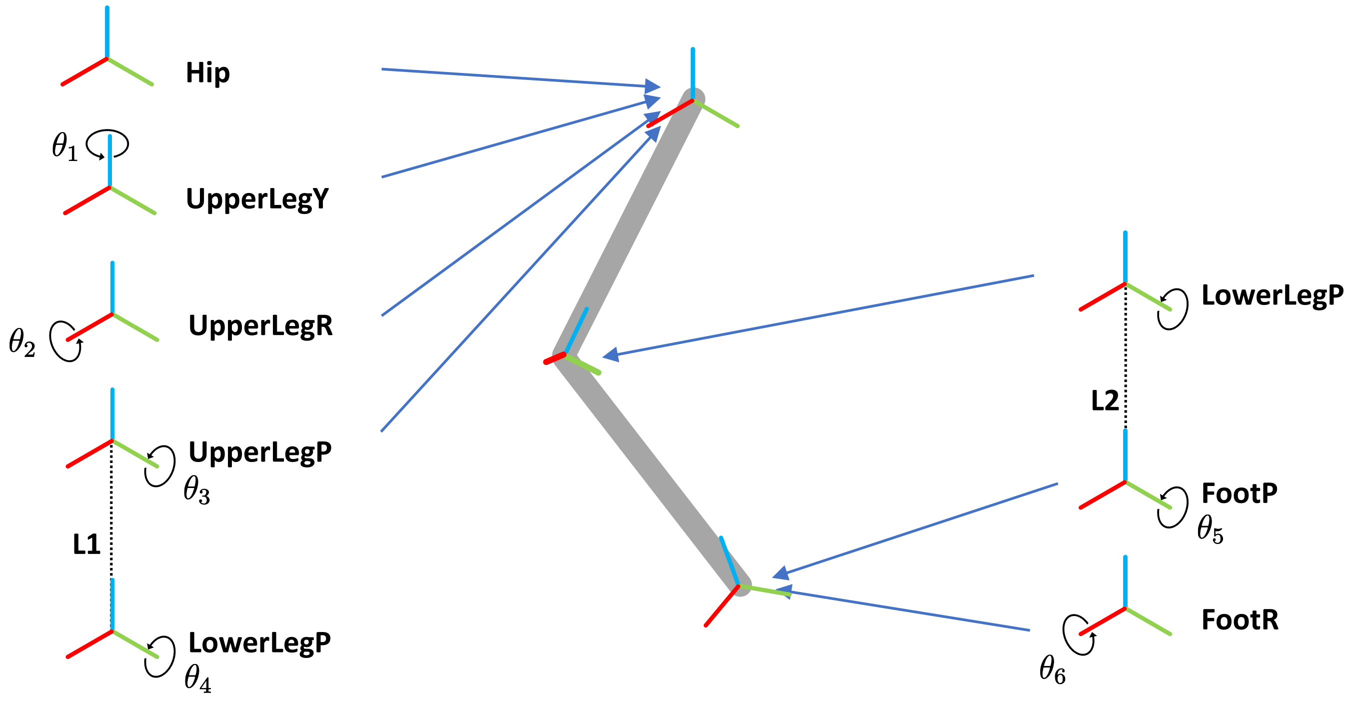
- Humanoid Robotics: A Reference
  - Differential Kinematics (Nenchev)
- Kajita’s book   An analytical solution to 6-DOF leg IK
- [Kofinas2013] Analytical IK of NAO
- [He2021] Analytical IK of Panda arm

Software

- Openrave [Diankov2010]
  - IKFast : an analytical IK solver for common kinematic chains

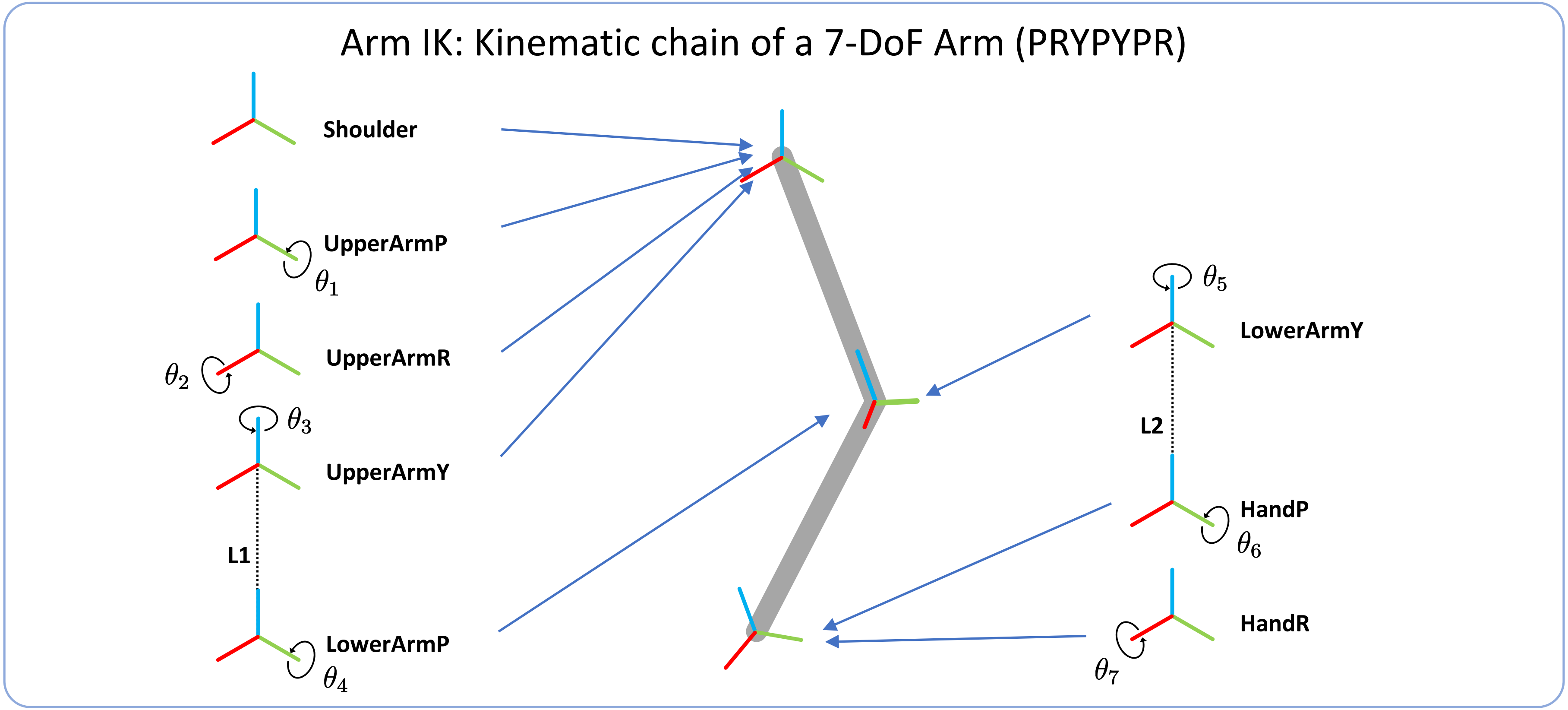
Kinematics

Leg IK: Kinematic chain of a 6-DoF leg (YRPPPR)



* Full derivation of leg and arm IK is provided in a document in vnoid repository.

Kinematics



Arm IK: Kinematic chain of a 7-DoF Arm (PRYPYPR)
Shoulder
UpperArmP
θ_1
UpperArmR
θ_2
θ_3
UpperArmY
L1
LowerArmP
θ_4
θ_5
LowerArmY
L2
HandP
θ_6
HandR
θ_7

Kinematics

IK can be calculated using trigonometry:

ex) knee and ankle pitch angles

$$\alpha = -\text{atan2}(p''_{\text{x}}, -p''_{\text{z}})$$

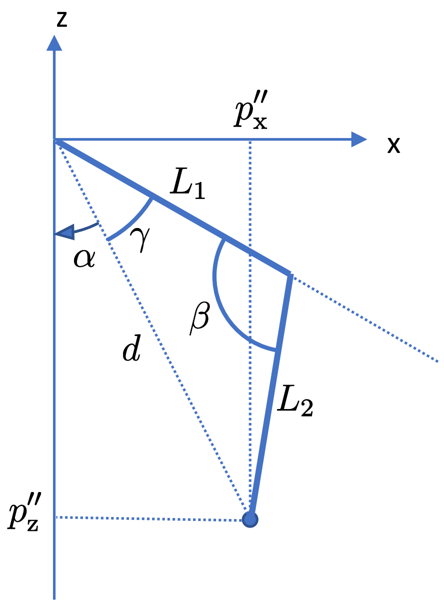
$$\beta = \text{acos}\left(\frac{L_1^2 + L_2^2 - d^2}{2L_1L_2}\right) \qquad d = \sqrt{p''^2_{\text{x}} + p''^2_{\text{z}}}$$

$$\gamma = \text{asin}\left(\frac{L_2\sin(\beta)}{d}\right)$$

$\theta_2 = \alpha - \gamma$ **hip pitch**

$\theta_3 = \pi - \beta$ **knee**

2D plane penetrating hip, knee, and ankle



Kinematics

IK can be calculated using trigonometry:

ex) knee and ankle pitch angles

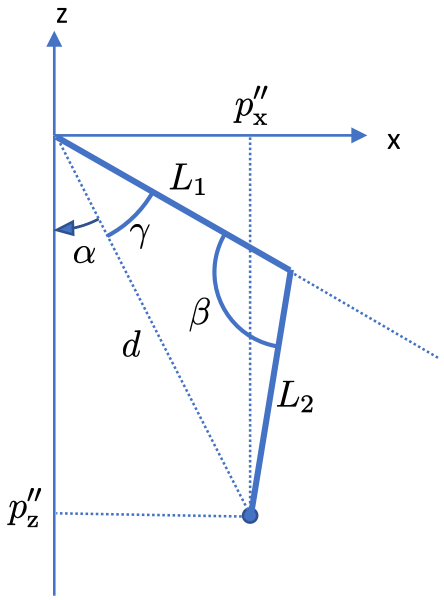
$$\beta = \mathrm{acos}\left(\frac{L_1^2 + L_2^2 - d^2}{2L_1L_2}\right)$$

< -1 => knee fully stretched

> +1 => knee fully bent

Note that we can easily detect singular postures by monitoring the argument of **acos**.
See the actual implementation for details.

2D plane penetrating hip, knee, and ankle



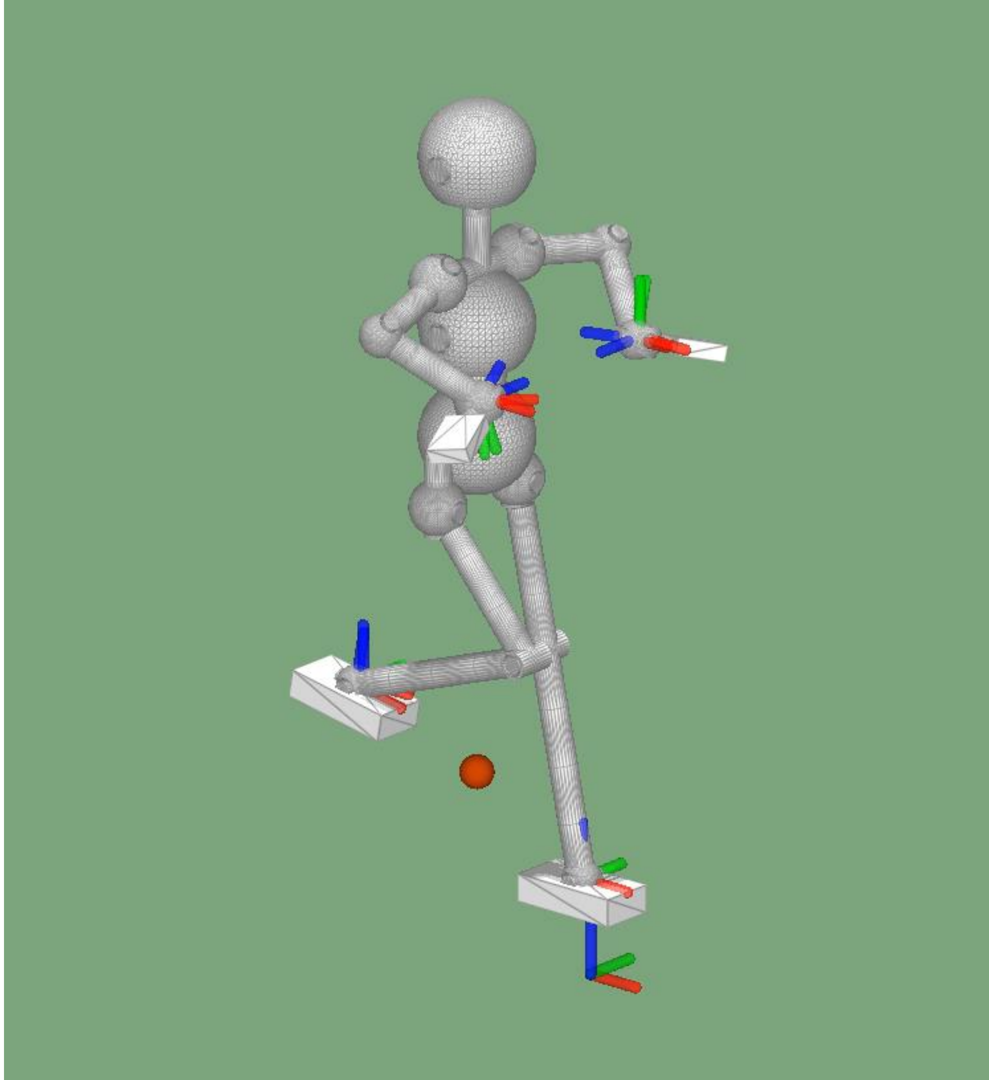
Kinematics

Simple IK demo 1

Desired hand and foot positions move along a circle

Knees and elbows fully stretch without problem (e.g., oscillation)

No knee buckling to the opposite side



Kinematics

Center-of-Mass IK

To compute a whole-body posture so that the **center-of-mass (CoM)** of the robot comes to a desired position.

Depends not only on kinematic but also kinetic information (i.e., mass distribution), the problem is called *inverse kinetics* in [Boulic].

Important IK problem for legged robots, because pattern generators and balance controllers output desired CoM position, not base link position.

Kinematics

Center-of-Mass IK

Once you have arm and leg IK and FK (forward kinetics) to compute the CoM of a given posture , you can implement CoM IK by simple iteration:

Input desired CoM position, desired Hand and Foot poses

Output base link position

base link position <- desired CoM position

Loop until error CoM position error is small enough:

    Compute **ArmIK** (desired arm pose, base link pose)

    Compute **LegIK** (desired leg pose, base link pose)

    Compute **CoMFK** (joint angles)

    Add CoM error to base link position

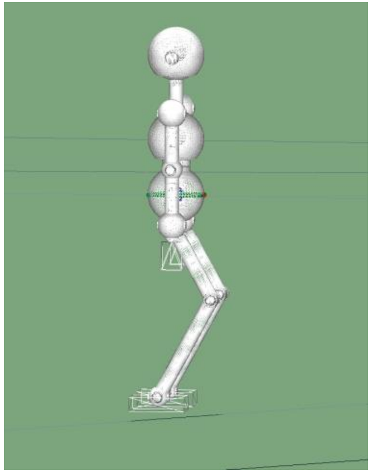
EndLoop

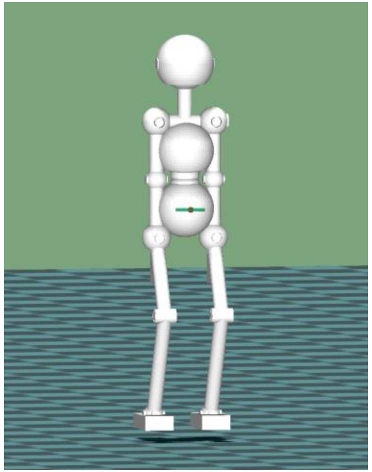
I think this idea is not new (it’s so simple)
But couldn’t identify a paper which proposed it first

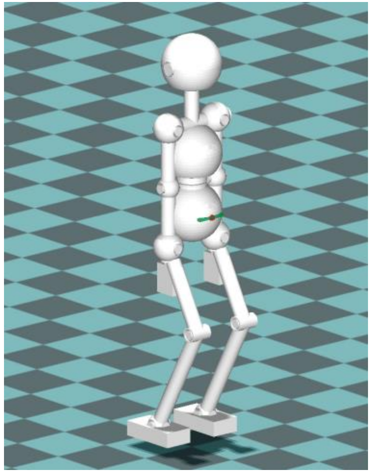
Kinematics

Simple IK demo 2

Desired hand and foot positions are fixed
Desired CoM position sways in x,y,z directions (amplitude 0.07m)







Reduced-order Models

Reduced-order models such as the **centroidal dynamics** and the **linear inverted pendulum mode** have been widely used for trajectory generation (walking pattern generation) and balance control.

By the way … the term **"template models"** has become quite popular, mainly in the context of machine learning (if I understand it right).

But they are not just templates in the sense of "(approximate) simplified dynamics".
Rather, **they can be rigorously derived from rigid-body equation without any approximation**.

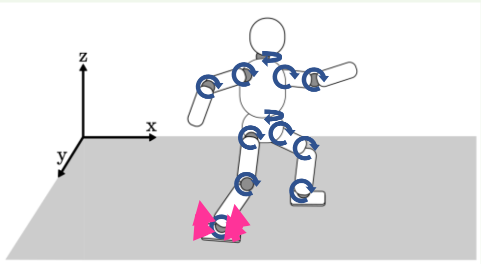
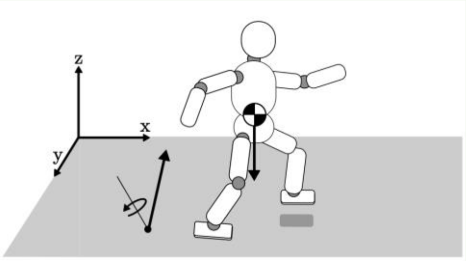
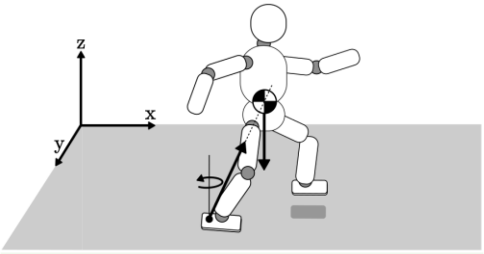
Reference to detailed derivation:

[Sugihara2018a] Derivation of centroidal dynamics from rigid-body equations
[Sugihara2018b] Derivation of LIPM from centroidal dynamics

Let us briefly review how these reduced-order models are derived from basic laws of mechanics.
(basically the same as Sugihara's derivation with slight difference in details)

Reduced-order Models

Comparison of different dynamical models

	Rigid-body dynamics (RBD)	**Centroidal dynamics (CD)**	**Linear inverted pendulum mode (LIPM)**
equation of motion	$M(\boldsymbol{q})\dot{\boldsymbol{v}} = h(\boldsymbol{q},\boldsymbol{v}) + g(\boldsymbol{q}) + J(\boldsymbol{q})^{\mathsf{T}}\boldsymbol{\lambda} + S^{\mathsf{T}}\boldsymbol{\tau}$	$m\ddot{\boldsymbol{p}}_{\mathrm{com}} = \boldsymbol{f} - m\boldsymbol{g}$ $\dot{\boldsymbol{L}} = \boldsymbol{\tau} + \boldsymbol{p}_{\mathrm{com}} \times \boldsymbol{f}$	$\ddot{\boldsymbol{p}}_{\mathrm{com}}(t) = \frac{1}{T^2}(\boldsymbol{p}_{\mathrm{com}}(t) - \boldsymbol{p}_{\mathrm{zmp}}(t)) - \boldsymbol{g}$
dimensionality (*not including contact forces)	**6 + num. of joints**	**6**	**3**
			

Derivation of Centroidal Dynamics

Consider a system of multiple rigid bodies.

m_i Mass of rigid body i

$\boldsymbol{p}_i$ CoM of rigid body i

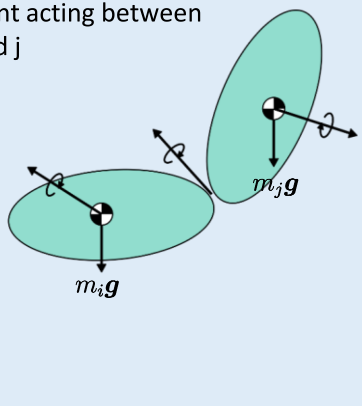
$\boldsymbol{L}_i$ Angular momentum of rigid body i

$\hat{\boldsymbol{f}}_i$ External force acting on rigid body i

$\hat{\boldsymbol{\tau}}_i$ External moment acting on rigid body i

$\boldsymbol{f}_{i,j}$ Internal force acting between rigid body i and j

$\boldsymbol{\tau}_{i,j}$ Internal moment acting between rigid body i and j



Equation of motion for each rigid body

$$m_i\ddot{\boldsymbol{p}}_i = \hat{\boldsymbol{f}}_i + \sum_j \boldsymbol{f}_{i,j} - m_i\boldsymbol{g}$$

$$\dot{\boldsymbol{L}}_i = \hat{\boldsymbol{\tau}}_i + \sum_j \boldsymbol{\tau}_{i,j}$$

Law of action and reaction

$$\boldsymbol{f}_{i,j} = -\boldsymbol{f}_{j,i}$$

$$\boldsymbol{\tau}_{i,j} + \boldsymbol{p}_i \times \boldsymbol{f}_{i,j} = -(\boldsymbol{\tau}_{j,i} + \boldsymbol{p}_j \times \boldsymbol{f}_{j,i})$$

Center-of-mass of the multi-body system

$$\boldsymbol{p}_{\text{com}} = \frac{1}{m}\sum_i m_i\boldsymbol{p}_i$$

Total angular momentum of the multi-body system around its CoM

$$\boldsymbol{L} = \sum_i (\boldsymbol{L}_i + (\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times m_i\dot{\boldsymbol{p}}_i)$$

Derivation of Centroidal Dynamics

Equation of motion for CoM

$$m\ddot{\boldsymbol{p}}_{\text{com}} = \sum_i m_i \ddot{\boldsymbol{p}}_i = \sum_i (\hat{\boldsymbol{f}}_i + \sum_j \boldsymbol{f}_{i,j} - m_i \boldsymbol{g}) = \sum_i \hat{\boldsymbol{f}}_i - m\boldsymbol{g}$$

$$m\ddot{\boldsymbol{p}}_{\text{com}} = \sum_i \hat{\boldsymbol{f}}_i - m\boldsymbol{g}$$

$$\dot{\boldsymbol{L}} = \sum_i (\hat{\boldsymbol{\tau}}_i + (\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times \hat{\boldsymbol{f}}_i)$$

All terms related internal forces vanish (verify yourself!)

Equation of motion for angular momentum

$$\begin{aligned}
\dot{\boldsymbol{L}} &= \sum_i (\dot{\boldsymbol{L}}_i + (\dot{\boldsymbol{p}}_i - \dot{\boldsymbol{p}}_{\text{com}}) \times m_i \dot{\boldsymbol{p}}_i + (\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times m_i \ddot{\boldsymbol{p}}_i) \\
&= \sum_i (\hat{\boldsymbol{\tau}}_i + \sum_j \boldsymbol{\tau}_{i,j} + \dot{\boldsymbol{p}}_i \times m_i \dot{\boldsymbol{p}}_i - \dot{\boldsymbol{p}}_{\text{com}} \times m_i \dot{\boldsymbol{p}}_i + (\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times (\hat{\boldsymbol{f}}_i + \sum_j \boldsymbol{f}_{i,j} - m_i \boldsymbol{g})) \\
&= \sum_i \hat{\boldsymbol{\tau}}_i + \sum_{i,j} \boldsymbol{\tau}_{i,j} + \sum_i (\dot{\boldsymbol{p}}_i \times m_i \dot{\boldsymbol{p}}_i) - \dot{\boldsymbol{p}}_{\text{com}} \times \sum_i m_i \dot{\boldsymbol{p}}_i + \sum_i ((\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times \hat{\boldsymbol{f}}_i) + \sum_i ((\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times \sum_j \boldsymbol{f}_{i,j}) - \sum_i ((\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times m_i \boldsymbol{g}) \\
&= \sum_i \hat{\boldsymbol{\tau}}_i + \sum_{i,j} (\boldsymbol{\tau}_{i,j} + \boldsymbol{\tau}_{j,i} + (\boldsymbol{p}_i - \boldsymbol{p}_j) \times \boldsymbol{f}_{i,j}) + \sum_i ((\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times \hat{\boldsymbol{f}}_i) - \boldsymbol{p}_{\text{com}} \times \sum_{i,j} \boldsymbol{f}_{i,j} - \sum_i m_i \boldsymbol{p}_i \times \boldsymbol{g} + \boldsymbol{p}_{\text{com}} \times \sum_i m_i \boldsymbol{g} \\
&= \sum_i (\hat{\boldsymbol{\tau}}_i + (\boldsymbol{p}_i - \boldsymbol{p}_{\text{com}}) \times \hat{\boldsymbol{f}}_i)
\end{aligned}$$

Derivation of Centroidal Dynamics

Now, let us define the **total contact wrench** as the total external force and moment acting on the rigid bodies.

Total contact wrench

$$\begin{bmatrix} \boldsymbol{f}_\mathrm{c} \\ \boldsymbol{\tau}_\mathrm{c} \end{bmatrix} = \begin{bmatrix} \sum_i \hat{\boldsymbol{f}}_i \\ \sum_i (\hat{\boldsymbol{\tau}}_i + \boldsymbol{p}_i \times \hat{\boldsymbol{f}}_i) \end{bmatrix}$$

Using the contact wrench, the equation of motion for the CoM and the total angular momentum of the multi-body system can be expressed as follows. This is called the **centroidal dynamics (CD)**.

Centroidal dynamics

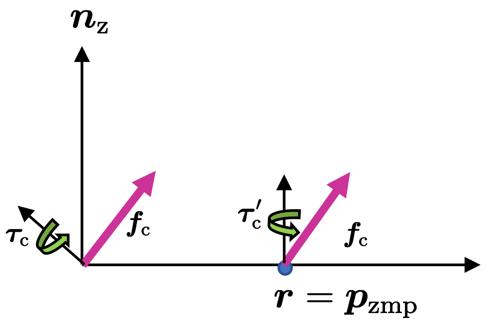
$$\begin{cases} m\ddot{\boldsymbol{p}}_\mathrm{com} = \boldsymbol{f}_\mathrm{c} - m\boldsymbol{g} \\ \dot{\boldsymbol{L}} = \boldsymbol{\tau}_\mathrm{c} - \boldsymbol{p}_\mathrm{com} \times \boldsymbol{f}_\mathrm{c} \end{cases}$$

Derivation of Linear Inverted Pendulum Mode

From contact wrench to **Zero-moment Point (ZMP)**

Pick an arbitrary point $\boldsymbol{r}$ on the ground plane and transform the contact wrench so that its point of application is $\boldsymbol{r}$.

$$\begin{bmatrix} \boldsymbol{f}_\mathrm{c} \\ \boldsymbol{\tau}_\mathrm{c} \end{bmatrix} \Rightarrow \begin{bmatrix} \boldsymbol{f}_\mathrm{c} \\ \boldsymbol{\tau}'_\mathrm{c} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_\mathrm{c} \\ \boldsymbol{\tau}_\mathrm{c} - \boldsymbol{r} \times \boldsymbol{f}_\mathrm{c} \end{bmatrix}$$



Now, consider a special $\boldsymbol{r}$ around which the <u>moment becomes perpendicular to the plane</u>.

$$\begin{aligned} &\boldsymbol{n}_\mathrm{z} \times \boldsymbol{\tau}'_\mathrm{c} \\ &= \boldsymbol{n}_\mathrm{z} \times (\boldsymbol{\tau}_\mathrm{c} - \boldsymbol{r} \times \boldsymbol{f}_\mathrm{c}) \\ &= \boldsymbol{n}_\mathrm{z} \times \boldsymbol{\tau}_\mathrm{c} - (\boldsymbol{n}_\mathrm{z} \cdot \boldsymbol{f}_\mathrm{c})\boldsymbol{r} + (\boldsymbol{n}_\mathrm{z} \cdot \boldsymbol{r})\boldsymbol{f}_\mathrm{c} \\ &= \boldsymbol{0} \end{aligned}$$

This point is the zero (tilting) moment point (ZMP).

ZMP (Zero tilting Moment Point) [Caron2017]

$$\boldsymbol{p}_\mathrm{zmp} = \frac{\boldsymbol{n}_\mathrm{z} \times \boldsymbol{\tau}_\mathrm{c}}{\boldsymbol{n}_\mathrm{z} \cdot \boldsymbol{f}_\mathrm{c}}$$

Derivation of Linear Inverted Pendulum Mode

Substitute the expression of the contact wrench around the ZMP to the centroidal equation of motion:

$$\begin{cases} m\ddot{\boldsymbol{p}}_{\mathrm{com}} = \boldsymbol{f}_{\mathrm{c}} - m\boldsymbol{g} \\ \dot{\boldsymbol{L}} = \boldsymbol{\tau}'_{\mathrm{c}} + (\boldsymbol{p}_{\mathrm{zmp}} - \boldsymbol{p}_{\mathrm{com}}) \times \boldsymbol{f}_{\mathrm{c}} \end{cases}$$

By eliminating $\boldsymbol{f}_{\mathrm{c}}$, we get

$$\dot{\boldsymbol{L}} = \boldsymbol{\tau}'_{\mathrm{c}} - m(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{zmp}}) \times (\ddot{\boldsymbol{p}}_{\mathrm{com}} + \boldsymbol{g})$$

We can come to this point by simple substitution.
What next? Where is LIPM?

The cross product becomes zero if and only if $\ddot{\boldsymbol{p}}_{\mathrm{com}} + \boldsymbol{g} \propto \boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{zmp}}$; that is, if

$$\ddot{\boldsymbol{p}}_{\mathrm{com}} = \frac{1}{T^2}(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{zmp}}) - \boldsymbol{g}$$

holds for some non-zero T.

This is the **linear inverted pendulum mode (LIPM)**.

Derivation of Linear Inverted Pendulum Mode

So what does this mean?

1. Linear and angular momentum are decoupled as long as linear CoM movement follows the LIPM.

$$\ddot{\boldsymbol{p}}_{\mathrm{com}} = \frac{1}{T^2}(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{zmp}}) - \boldsymbol{g}$$

$$\dot{\boldsymbol{L}} = \boldsymbol{\tau}'_{\mathrm{c}} = \begin{bmatrix} 0 \\ 0 \\ \tau_{\mathrm{z}} \end{bmatrix}$$

This characteristic is utilized in walking pattern generation

2. You can manipulate the angular momentum by letting the linear CoM movement deviate from the LIPM.

$$\dot{\boldsymbol{L}} = \boldsymbol{\tau}'_{\mathrm{c}} - m(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{zmp}}) \times (\ddot{\boldsymbol{p}}_{\mathrm{com}} + \boldsymbol{g})$$

This characteristic is utilized in balance control strategy (some examples in next slide)

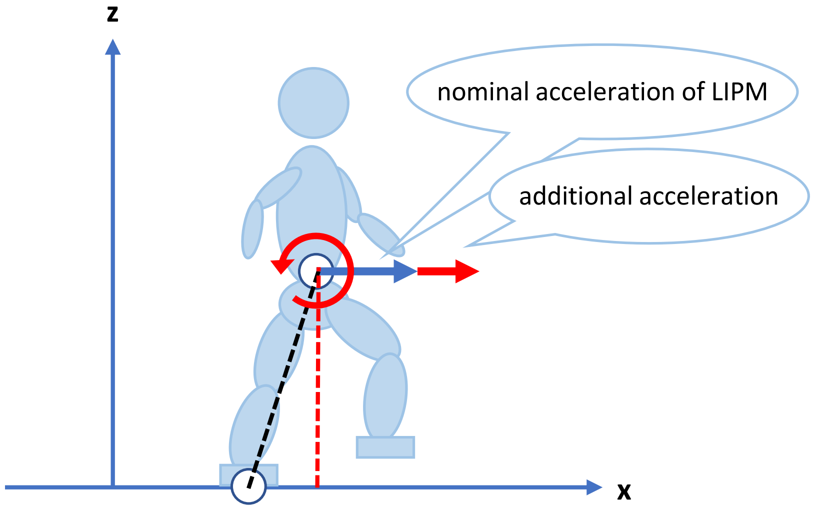
Important difference between CD and LIPM

CD is a law of physics. You can never violate it.

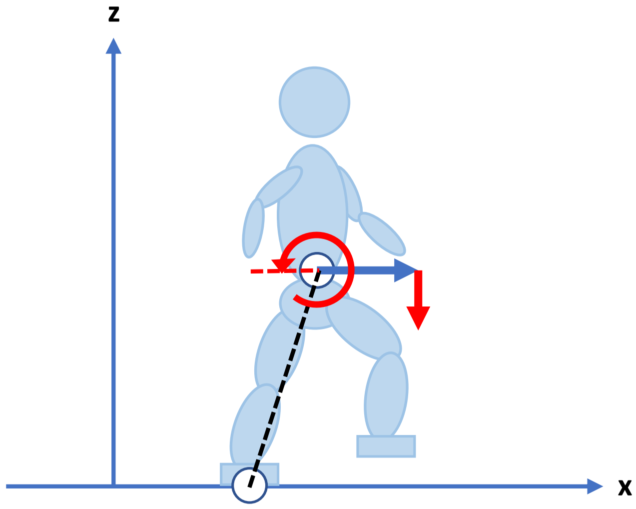
LIPM is an embedded subsystem (a mode) of CD.
You can violate it, but then the angular momentum will change due to coupling effect.

Derivation of Linear Inverted Pendulum Mode

CoM acceleration more or less than LIPM generates angular momentum.



Additional horizontal acceleration generates momentum



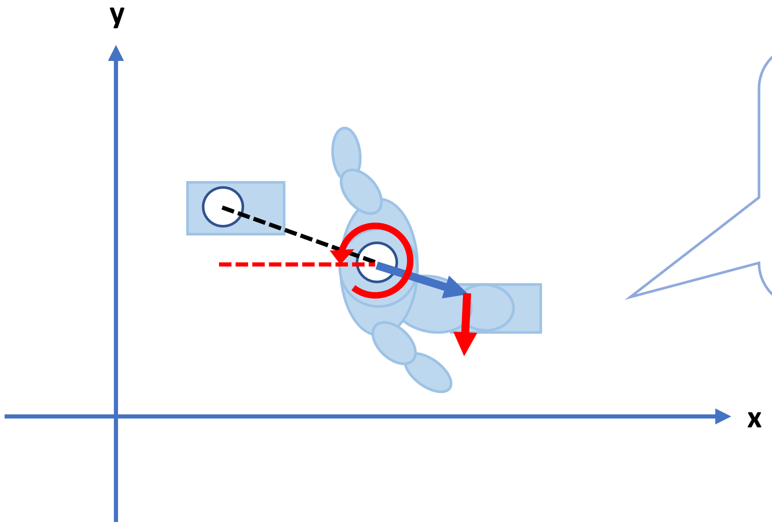
You can even use vertical acceleration.

In this case, there must be certain horizontal offset between the CoM and ZMP.

Derivation of Linear Inverted Pendulum Mode

CoM acceleration more or less than LIPM generates angular momentum.

You can also utilize this relationship for generating spinning (i.e., turning along z axis) moment without relying on contact moment from the ground.



Additional sideways acceleration used for generating yaw (turning) momentum.

Also in this case, there must be certain horizontal offset between the CoM and ZMP.

DCM Dynamics

Divergent Component of Motion (DCM)

The DCM of the LIPM is defined as

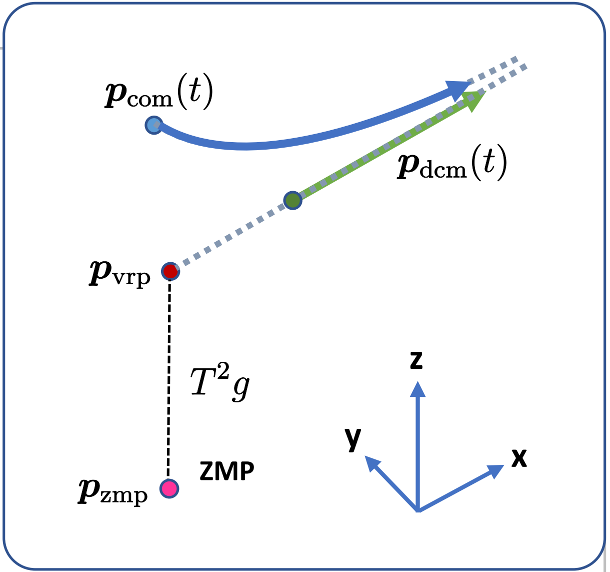
$$\boldsymbol{p}_{\mathrm{dcm}} = \boldsymbol{p}_{\mathrm{com}} + T\dot{\boldsymbol{p}}_{\mathrm{com}}$$

Using the DCM, the LIPM (a second-order ODE) can be decomposed in to **a pair of first-order, stable and unstable dynamics**.

$$\dot{\boldsymbol{p}}_{\mathrm{com}} = -\frac{1}{T}(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{dcm}})$$

$$\dot{\boldsymbol{p}}_{\mathrm{dcm}} = \frac{1}{T}(\boldsymbol{p}_{\mathrm{dcm}} - \boldsymbol{p}_{\mathrm{vrp}})$$

virtual repellent point (VRP)



This means that the **DCM moves away from the VRP.**
If VRP is a fixed point, then DCM moves along a straight line.

Moreover, the **CoM follows DCM** thanks to its stable dynamics.

DCM Dynamics

Relationship between the angular momentum and the DCM dynamics

Recall that the following holds.

$$\dot{\boldsymbol{L}} = \boldsymbol{\tau}_{\mathrm{c}}' - m(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{zmp}}) \times (\ddot{\boldsymbol{p}}_{\mathrm{com}} + \boldsymbol{g})$$

We can also write

$$\dot{\boldsymbol{L}} = \boldsymbol{\tau}_{\mathrm{c}}' - m(\boldsymbol{p}_{\mathrm{com}} - \boldsymbol{p}_{\mathrm{zmp}}) \times \frac{1}{T}\left(\dot{\boldsymbol{p}}_{\mathrm{dcm}} - \frac{1}{T}(\boldsymbol{p}_{\mathrm{dcm}} - \boldsymbol{p}_{\mathrm{vrp}})\right)$$

which means that the **deviation from the DCM dynamics generates angular momentum**.

Balance Control

Review of Existing Stabilizers

Name	Description	Measure/Estimate
Torso position compliance control [Nagasaka 1999a,1999b]	Adjust desired CoM to track LIPM Manipulate ZMP to regulate torso inclination	Base link tilt
ZMP preview control [Kajita 2003]	Manipulate desired CoM jerk to stabilize CoM and track ZMP to reference	CoM state
Model ZMP control [Takenaka 2009,2015]	Adjust desired CoM acceleration to generate recovery moment	Base link tilt
DCM tracking [Englsberger 2013]	Manipulate VRP to regulate DCM tracking error.	DCM
DCM-based step adjustment [Khadiv 2016]	Adjust landing position and step duration to have desired DCM offset at landing	DCM
Foot-guided agile control [Sugihara 2017]	Manipulate ZMP so that DCM at landing matches the foot landing position.	CoM state
[Kojio 2019,2020]	Extention of [Sugihara2017] to incorporate landing adaptation	

Let us formulate a balance control strategy similar to [Nagasaka] and [Takenaka] based on the relationship of DCM dynamics and angular momentum.

Balance Control

Recall again the following.

$$\dot{\boldsymbol{L}} = \boldsymbol{\tau}_c' - m(\boldsymbol{p}_{com} - \boldsymbol{p}_{zmp}) \times \frac{1}{T}\left(\dot{\boldsymbol{p}}_{dcm} - \frac{1}{T}(\boldsymbol{p}_{dcm} - \boldsymbol{p}_{vrp})\right)$$

For simplicity, let us assume that the CoM height is given by a constant h .
Then we can write

$$\dot{L}_x = \frac{mh}{T}\left(\dot{p}_{dcm,y} - \frac{1}{T}(p_{dcm,y} - p_{vrp,y})\right)$$

$$\dot{L}_y = -\frac{mh}{T}\left(\dot{p}_{dcm,x} - \frac{1}{T}(p_{dcm,x} - p_{vrp,x})\right)$$

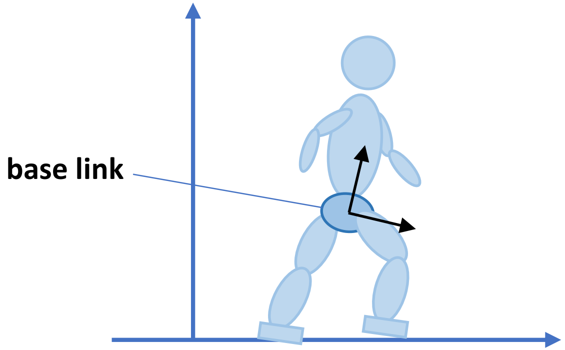
Deviation from the DCM dynamics generates change of angular momentum.

Of course, we have an equation for Lz, but let us ignore it…

$$\dot{L}_z = \frac{m}{T}\left(-(p_{com,x} - p_{zmp,x})\left(\dot{p}_{dcm,y} - \frac{1}{T}(p_{dcm,y} - p_{vrp,y})\right) + (p_{com,y} - p_{zmp,y})\left(\dot{p}_{dcm,x} - \frac{1}{T}(p_{dcm,x} - p_{vrp,x})\right)\right)$$

Balance Control

Let $\theta_{x,y}$ and $\omega_{x,y}$ be the tilting angle and angular velocity of the base link.
These can be measured using an IMU (or a RateGyro sensor of Choreonoid).



We would like to regulate $\theta_{x,y}$ and $\omega_{x,y}$ to maintain the orientation of the base link.
To realize this, consider the following simple PD law.

$$\dot{L}_{x}^{d} = -(K_{\theta}\theta_{x} + K_{\omega}\omega_{x})$$
$$\dot{L}_{y}^{d} = -(K_{\theta}\theta_{y} + K_{\omega}\omega_{y})$$

Desired moment to recover balance

Balance Control

Now, the balance recovery moment is input to the DCM dynamics as a disturbance:

$$\dot{p}_{\mathrm{dcm,x}} = \frac{1}{T}(p_{\mathrm{dcm,x}} - p_{\mathrm{vrp,x}}) - \frac{T}{mh}\dot{L}_{\mathrm{y}}^{\mathrm{d}}$$
$$\dot{p}_{\mathrm{dcm,y}} = \frac{1}{T}(p_{\mathrm{dcm,y}} - p_{\mathrm{vrp,y}}) + \frac{T}{mh}\dot{L}_{\mathrm{x}}^{\mathrm{d}}$$

Deviation from the nominal DCM dynamics generates recovery moment.

Since the DCM dynamics is inherently unstable, we need to stabilize it.
We manipulate ZMP (VRP) to stabilize the DCM.

$$\boldsymbol{p}_{\mathrm{vrp}} = \boldsymbol{p}_{\mathrm{vrp}}^{\mathrm{ref}} - K_{\mathrm{dcm}}(\boldsymbol{p}_{\mathrm{dcm}} - \boldsymbol{p}_{\mathrm{dcm}}^{\mathrm{ref}})$$

VRP (i.e., ZMP) is used to control the DCM to track a reference trajectory.

desired VRP

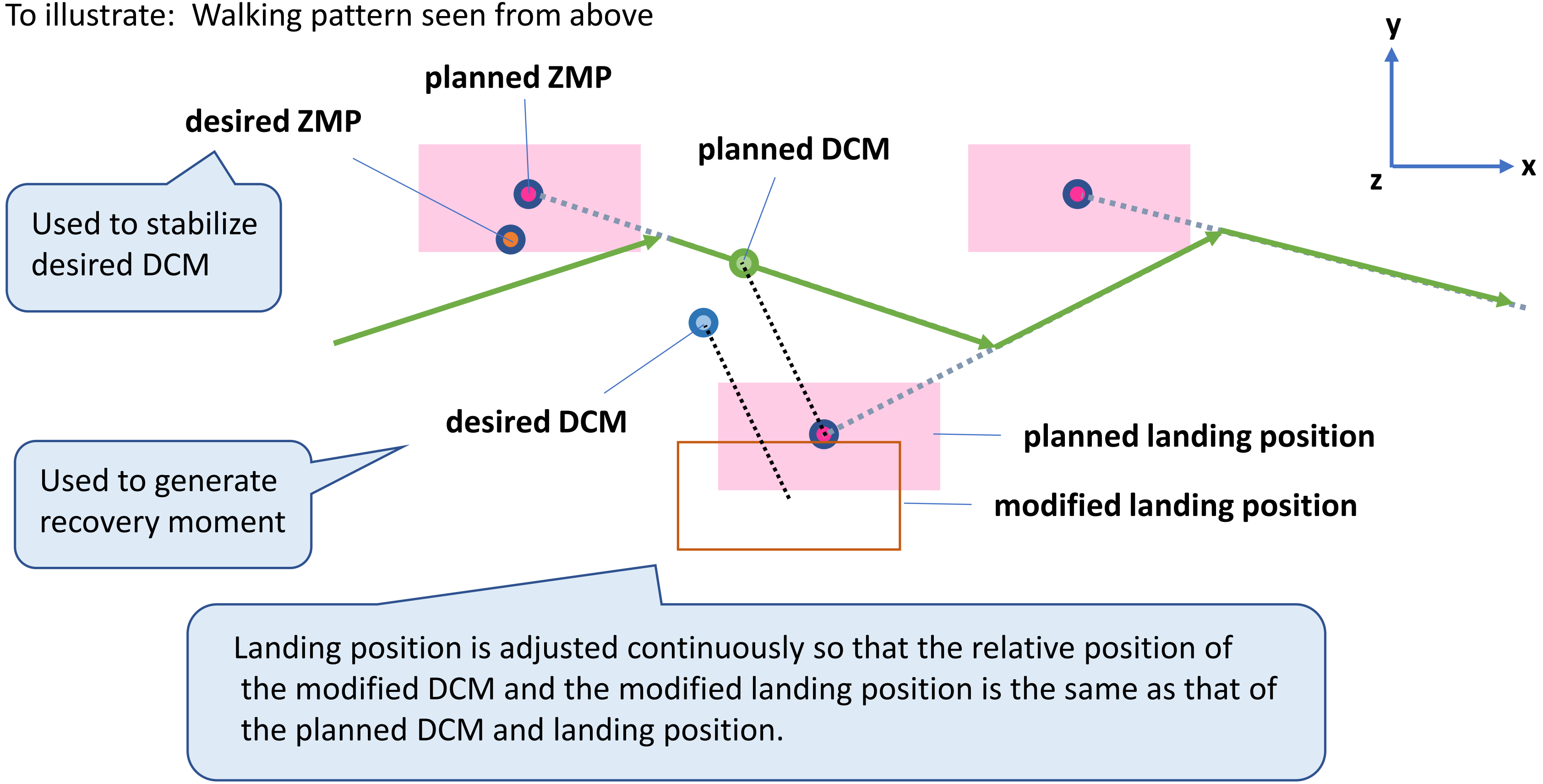
planned VRP

desired DCM

planned DCM

Balance Control

To illustrate: Walking pattern seen from above



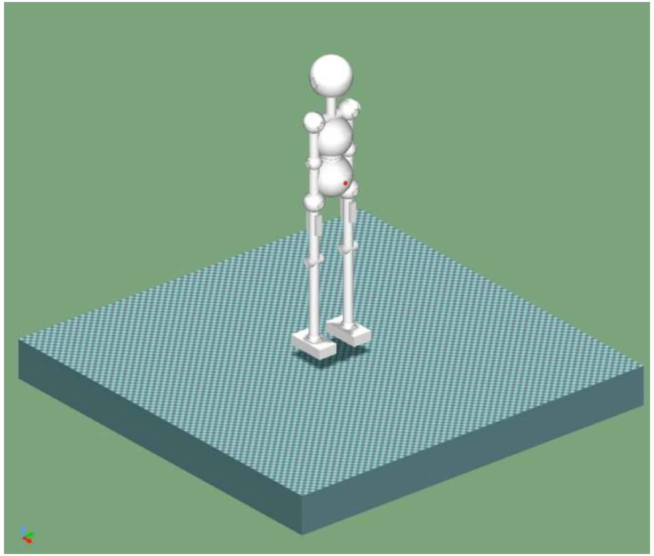
Examples

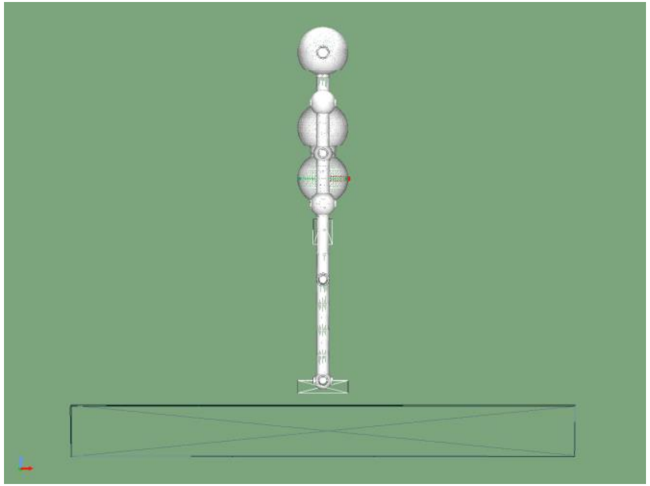
Simple balance control demo 1

The floor moves horizontally in a sinusoidal pattern

Amplitide: 0.2 [m]
Frequency: 2.0 [rad/s]

Red marker: desired ZMP
Green marker: actual ZMP



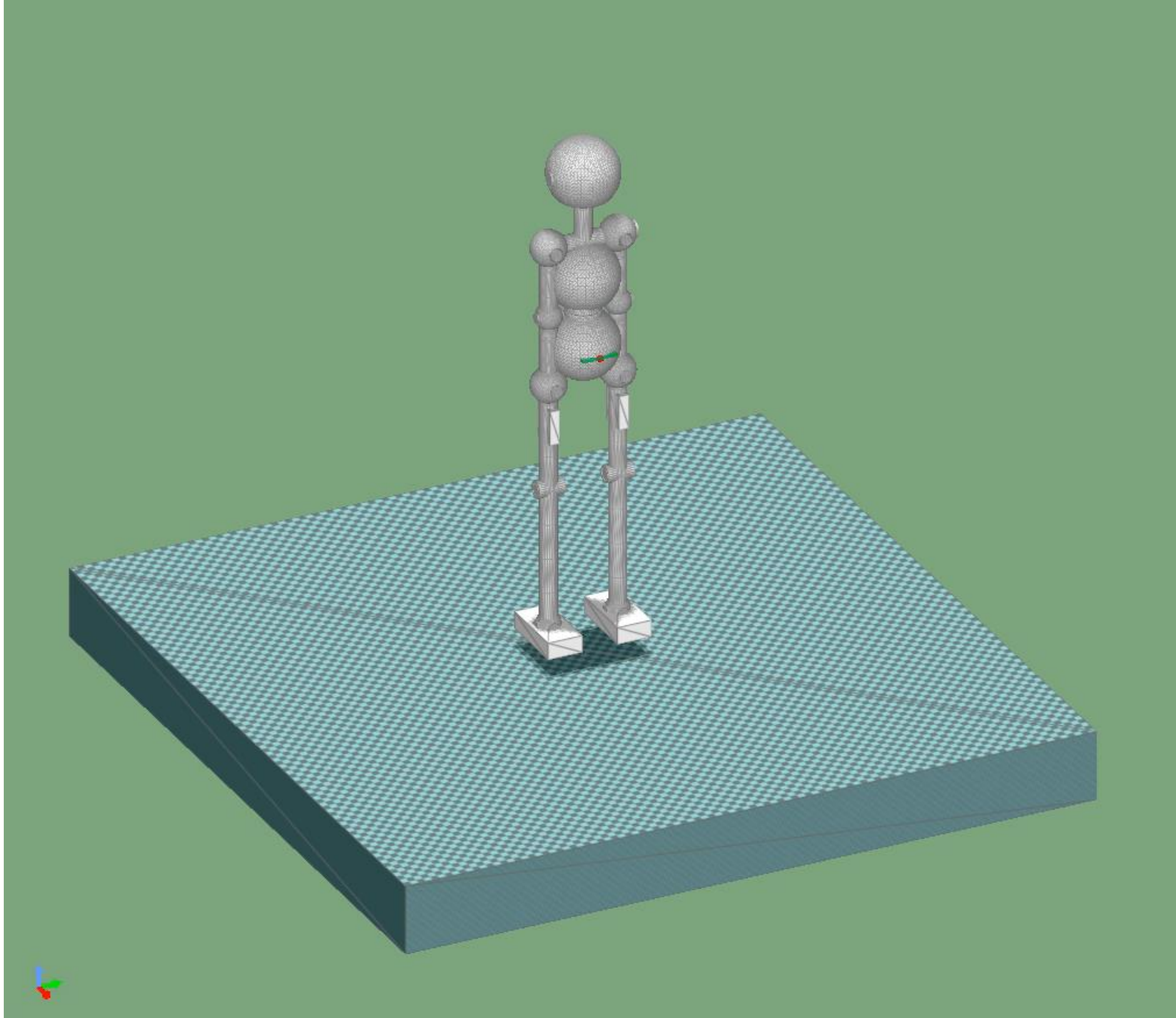


Examples

Simple balance control demo 3

The floor moves horizontally in a sinusoidal pattern

Amplitide: 0.2 [m]
Frequency: 2.0 [rad/s]



Examples

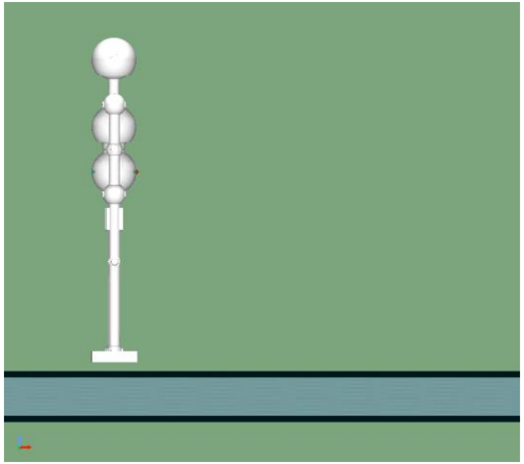
Simple balance control demo 4

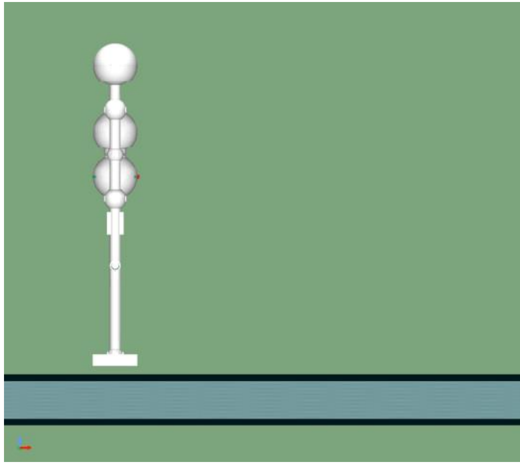
Impulsive disturbances while walking forward

20Ns (0.1m delta DCM)

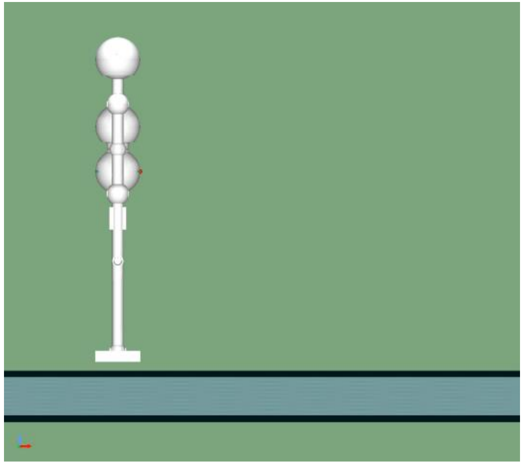
30Ns (0.15m delta DCM)

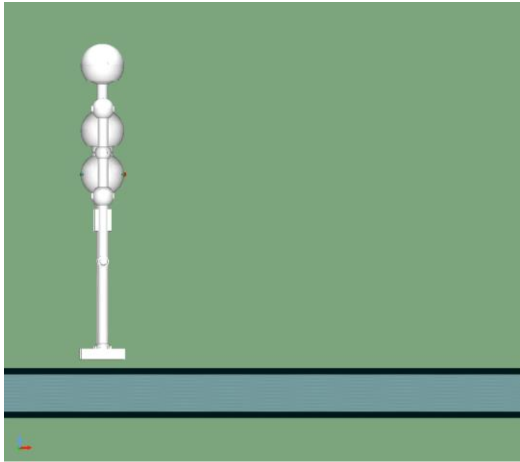
ZMP only





With DCM adjustment + step adaptation

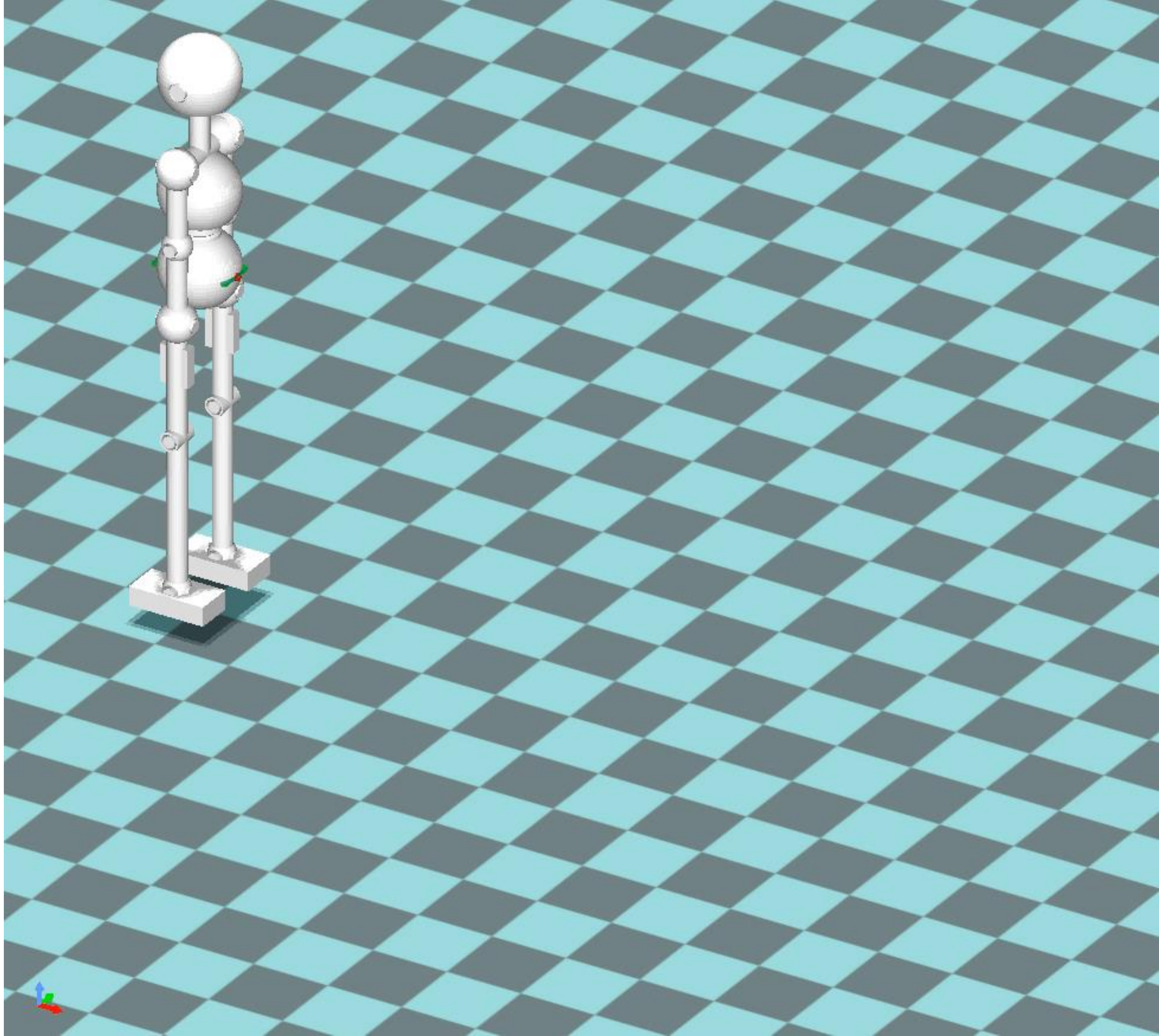




Examples

What happens with stronger disturbance…

When the body starts spinning, everything goes unpredictable…



To Conclude…

Analytical IK

Fast computation

Robust against singularities

Reduced-order models

Can be derived from rigid-body equations without approximation

DCM-based walking pattern generation and balance control

A simple framework for realizing robust walking based on ZMP, DCM, and step adaptation.

What if we want to do more?

Exploit full kinematics and whole-body dynamics?

Plan and regulate angular momentum for faster motion including flight phase?

Generate motions without pre-planned contact?

Beyond the scope of this talk
(but maybe within the scope of other talks of this tutorial!)

References

Kinematics

R. Boulic, R. Mas, D. Thalmann: “Inverse Kinetics for Center of Mass Position Control and Posture Optimization”, 1995.

S. Kajita: Humanoid Robot (梶田:ヒューマノイドロボット(改訂2版)), オーム社, 2020

Diankov: “Automated construction of robotic manipulation programs”, PhD Thesis, 2010.

N. Kofinas et al.: "Complete analytical inverse kinematics for NAO," 2013 13th International Conference on Autonomous Robot Systems, 2013, pp. 1-6.

Y. He and S. Liu: "Analytical Inverse Kinematics for Franka Emika Panda – a Geometrical Solver for 7-DOF Manipulators with Unconventional Design," ICCMA 2021.

Reduced-order models

T. Sugihara and Y. Fujimoto: “Dynamics Analysis: Equation of Motion”, Humanoid Robotics: A Reference, pp.1-32, 2017.

T. Sugihara and K. Yamane: “Reduced-Order Models”, Humanoid Robotics: A Reference, pp.1-38, 2017.

S. Caron, Q. -C. Pham and Y. Nakamura, "ZMP Support Areas for Multicontact Mobility Under Frictional Constraints," IEEE Transactions on Robotics, vol. 33, no. 1, pp. 67-80, 2017

Walking pattern generation

J. Englsberger, C. Ott and A. Albu-Schäffer, "Three-dimensional bipedal walking control using Divergent Component of Motion," IROS2013, pp. 2600-2607, 2013.

Stabilization

Nagasaka, Inaba, Inoue: Stabilization of Dynamic Walk on a Humanoid Using Torso Position Compliance Control, Annual Conference of RSJ, 1999. (in Japanese)

K.Nagasaka, Whole-body Motion Generation for a Humanoid Robot by Dyanmics Filters (Japanese), PhD Thesis, Univ. Tokyo, 1999. (in Japanese)

T. Takenaka, T. Matsumoto and T. Yoshiike, "Real time motion generation and control for biped robot -3rd report: Dynamics error compensation-," IROS2009, pp. 1594-1600, 2009.

T. Takenaka: PhD Thesis (竹中 2脚ロボットの歩行・走行歩容生成と姿勢制御) (in Japanese)

M. Khadiv, A. Herzog, S. A. A. Moosavian and L. Righetti, "Step timing adjustment: A step toward generating robust gaits," Humanoids2016, pp. 35-42, 2016.

T. Sugihara and T. Yamamoto: Foot-guided agile control of a biped robot through zmp manipulation. IROS2017, pp. 4546–4551, 2017.

Y. Kojio et al., "Footstep Modification Including Step Time and Angular Momentum Under Disturbances on Sparse Footholds," IEEE RAL, vol. 5, no. 3, pp. 4907-4914, 2020.