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
- much 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 vs Analytical solutions

	Pros	Cons
Numerical	Flexible (kinematic closed-loop, constraint-based, 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)

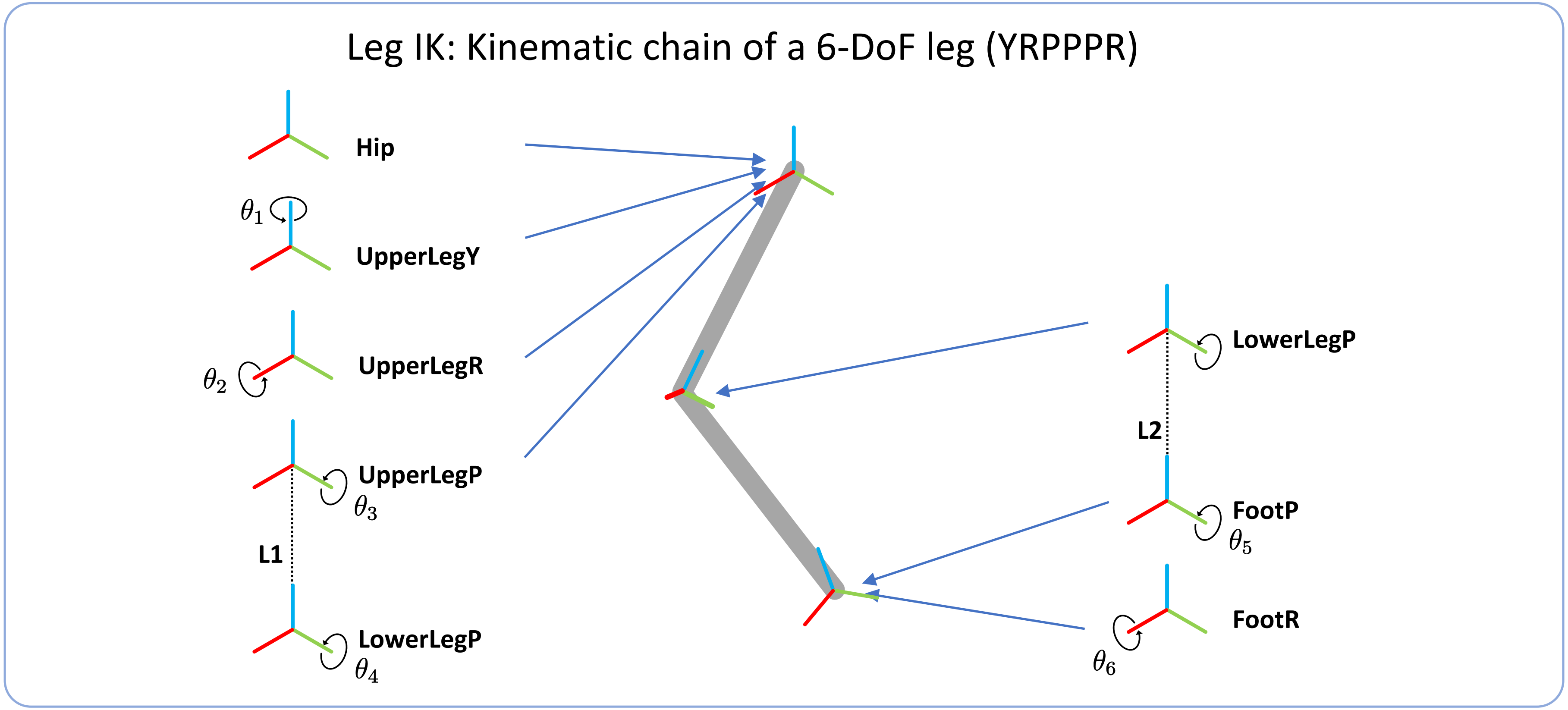
Kajita’s book   An analytical solution to 6-DOF leg IK

Software

Openrave (Diankov)

IKFast : a general purpose analytical IK solver

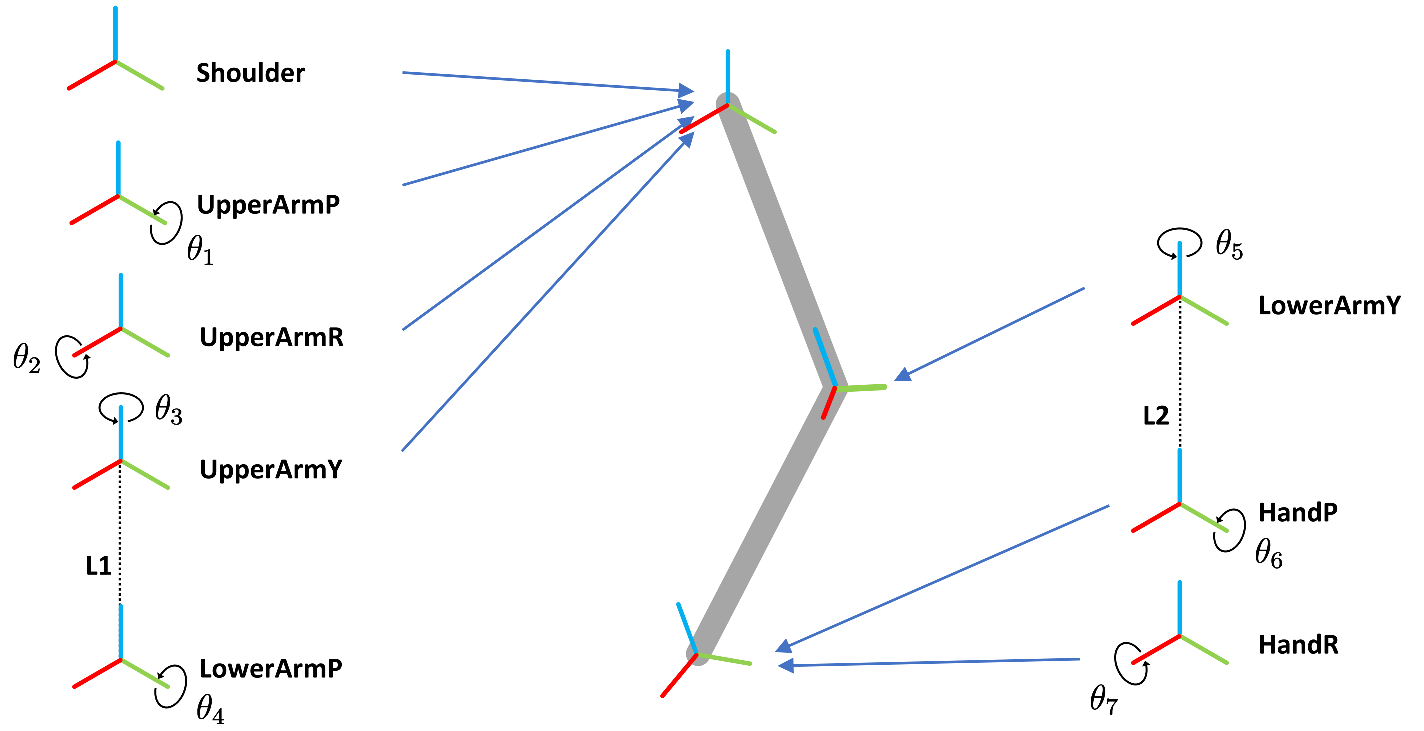
Kinematics



* Full derivation of leg and arm IK is provided in the appendix.

Kinematics

Arm IK: Kinematic chain of a 7-DoF Arm (PRYPYPR)



* Full derivation of leg and arm IK is provided in the appendix.

Kinematics

IK can be calculated using trigonometry:

ex) knee and ankle pitch angles

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

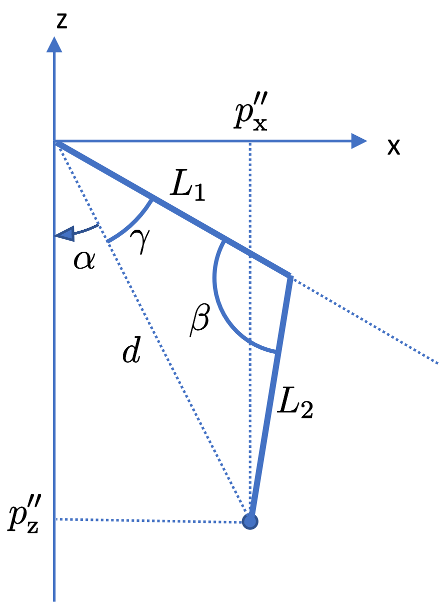
$$\beta = \mathrm{acos}\left(\frac{L_1^2 + L_2^2 - d^2}{2L_1L_2}\right) \qquad d = \sqrt{p''^2_{\mathrm{x}} + p''^2_{\mathrm{z}}}$$

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

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

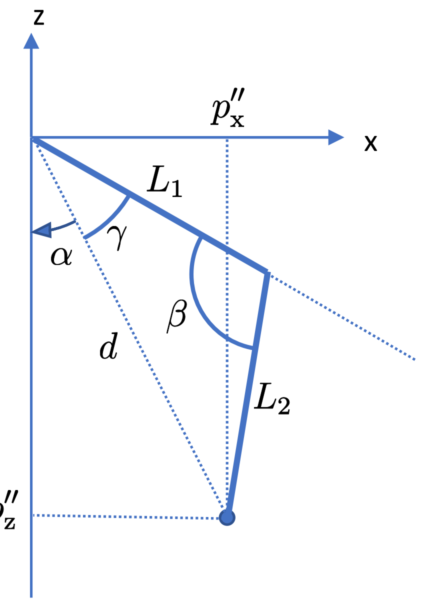
$$\beta = \mathrm{acos}\left(\frac{L_1^2 + L_2^2 - d^2}{2L_1L_2}\right)$$

< -1 => knee fully stretched

> +1 => knee fully bent

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

2D plane penetrating hip, knee, and ankle

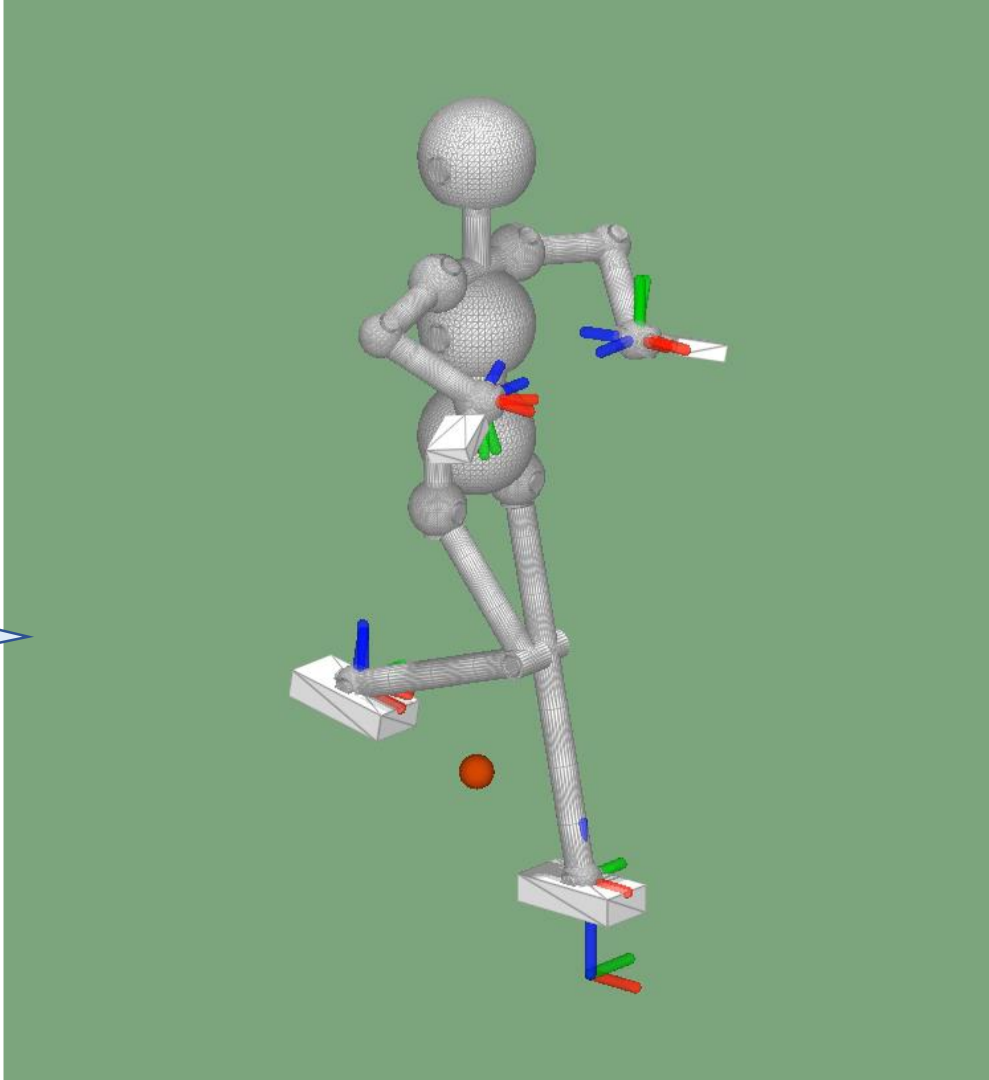


Kinematics

Leg and Arm IK demo using vnoid

Knee/elbow fully stretches without problem

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:

```
Init base link position with 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
```

Kinematics

Center-of-Mass IK

Demo of vnoid

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.

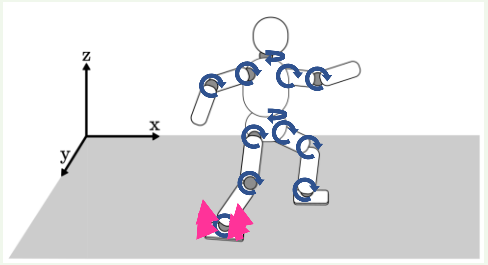
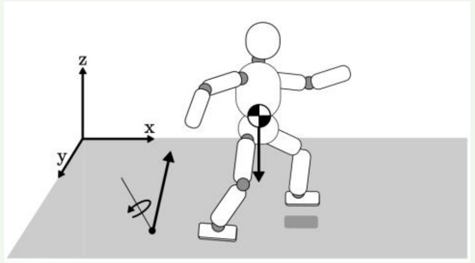
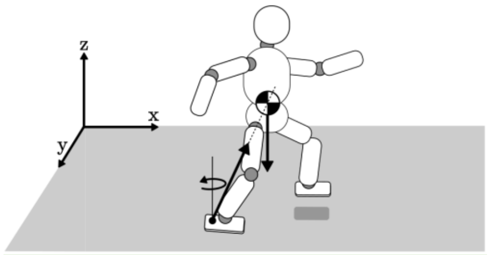
By the way … the term **"template models"** has become quite popular, mainly in the context of machine learning (if I understand it right).

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

So, before going into how to use them, let us review how they are derived from basic laws of mechanics.

Reduced-order Models

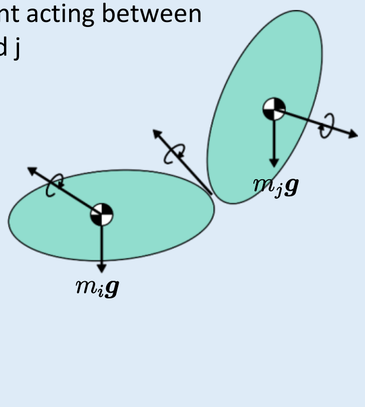
Comparison of different dynamical models

	Rigid-body dynamics	Centroidal dynamics	Linear inverted pendulum mode
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

Let us start from **multi-body dynamics**.

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

$$
\begin{aligned}
m\ddot{\boldsymbol{p}}_{\mathrm{com}} &= \sum_i m_i \ddot{\boldsymbol{p}}_i \\
&= \sum_i \Big(\hat{\boldsymbol{f}}_i + \sum_j \boldsymbol{f}_{i,j} - m_i \boldsymbol{g}\Big) \\
&= \underline{\sum_i \hat{\boldsymbol{f}}_i - m\boldsymbol{g}}
\end{aligned}
$$

$$
\begin{aligned}
m\ddot{\boldsymbol{p}}_{\mathrm{com}} &= \sum_i \hat{\boldsymbol{f}}_i - m\boldsymbol{g} \\
\dot{\boldsymbol{L}} &= \sum_i \big(\hat{\boldsymbol{\tau}}_i + (\boldsymbol{p}_i - \boldsymbol{p}_{\mathrm{com}}) \times \hat{\boldsymbol{f}}_i\big)
\end{aligned}
$$

Equation of motion for angular momentum

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

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

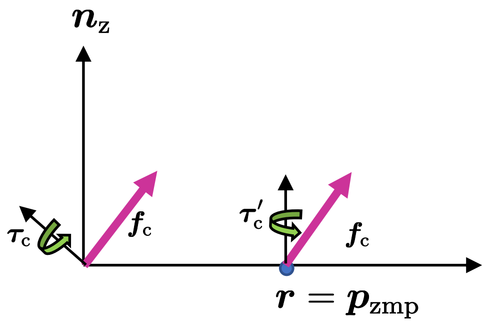
$$\begin{cases} m\ddot{\boldsymbol{p}}_{\mathrm{com}} = \boldsymbol{f}_{\mathrm{c}} - m\boldsymbol{g} \\ \dot{\boldsymbol{L}} = \boldsymbol{\tau}_{\mathrm{c}} - \boldsymbol{p}_{\mathrm{com}} \times \boldsymbol{f}_{\mathrm{c}} \end{cases}$$

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 $\boldsymbol{r}$.

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



$\boldsymbol{n}_\mathrm{z}$
$\boldsymbol{\tau}_\mathrm{c}$
$\boldsymbol{f}_\mathrm{c}$
$\boldsymbol{\tau}'_\mathrm{c}$
$\boldsymbol{f}_\mathrm{c}$
$\boldsymbol{r} = \boldsymbol{p}_\mathrm{zmp}$

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

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

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

ZMP (Zero tilting Moment Point)

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

Derivation of Linear Inverted Pendulum Mode

Substitude 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

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

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?

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

$$\ddot{\boldsymbol{p}}_{\text{com}} = \frac{1}{T^2}(\boldsymbol{p}_{\text{com}} - \boldsymbol{p}_{\text{zmp}}) - \boldsymbol{g}$$
$$\dot{\boldsymbol{L}} = \boldsymbol{\tau}_{\text{c}}'$$

In other words, you can manipulate the angular momentum by deviating the linear CoM movement from the LIPM.

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

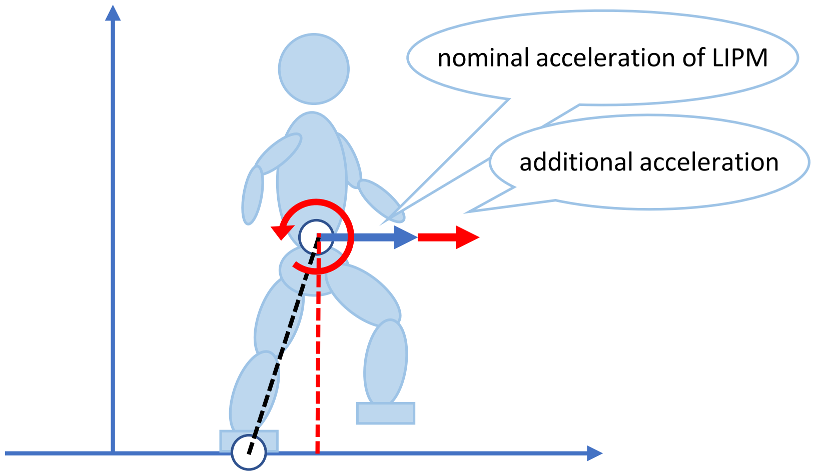
Important difference between CD and LIPM

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

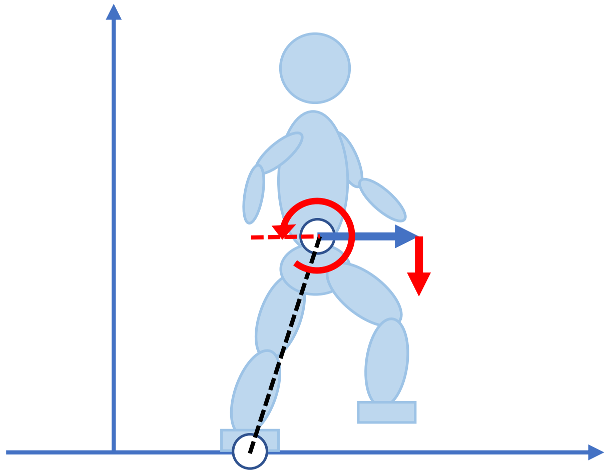
LIPM is an embedded subsystem (a mode) of CD.
You can violate it, but then the angular momentum will change due to coupling effect.

Derivation of Linear Inverted Pendulum Mode

How you can utilize this relationship to design balance control:



Additional horizontal acceleration generates momentum

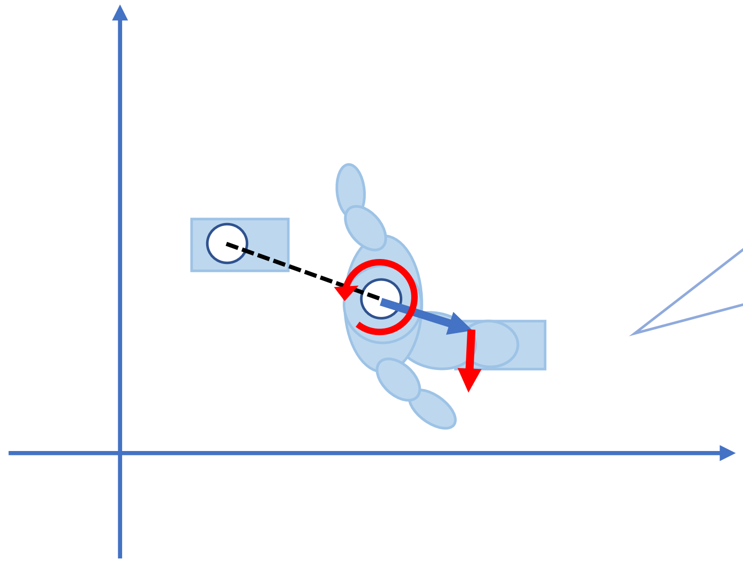


You can even use vertical acceleration.

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

Derivation of Linear Inverted Pendulum Mode

How you can utilize this relationship for quick turning (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.

Derivation of Linear Inverted Pendulum Mode

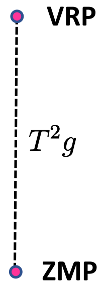
Virtual Repellent Point (VRP)

Let us define the virtual repellent point (VRP) as

$$\boldsymbol{p}_{\mathrm{vrp}} = \boldsymbol{p}_{\mathrm{zmp}} + T^2 \boldsymbol{g}$$

Then the LIPM can be written simply as

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

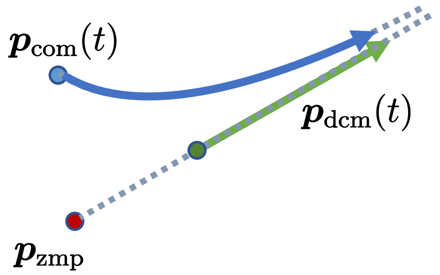


Derivation of Linear Inverted Pendulum Mode

Divergent Component of Motion (DCM)

The DCM of the LIPM is defined as

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



$\boldsymbol{p}_{\mathrm{com}}(t)$
$\boldsymbol{p}_{\mathrm{dcm}}(t)$
$\boldsymbol{p}_{\mathrm{zmp}}$

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

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

Now, consider that the ZMP is fixed during $[0, t]$; then by solving the ODE, we get

$$\boldsymbol{p}_{\mathrm{dcm}}(t) = \boldsymbol{p}_{\mathrm{zmp}} + \exp\left(\frac{t}{T}\right)(\boldsymbol{p}_{\mathrm{dcm}}(0) - \boldsymbol{p}_{\mathrm{zmp}})$$

Geometrically, this means that the **DCM moves away from the ZMP on a straight line**.

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

Derivation of Linear Inverted Pendulum Mode

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

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

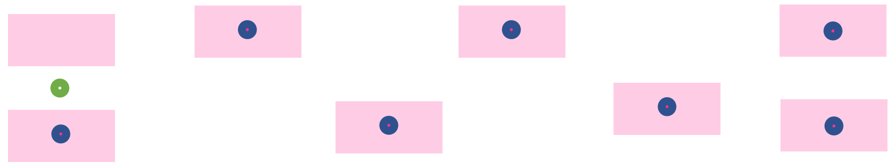
which means that the **deviation from the DCM dynamics generates angular momentum**.

Walking Pattern Generation [Englsberger 2013]

1. Determine footsteps, ZMPs, and step durations.

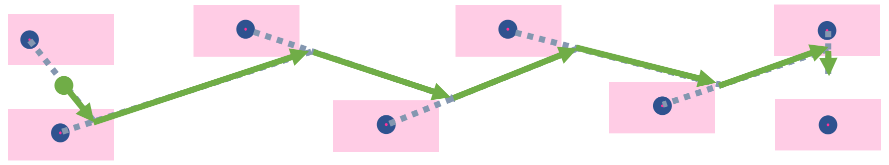
*the initial DCM is given.

*the ZMP of the first step is not determined yet.



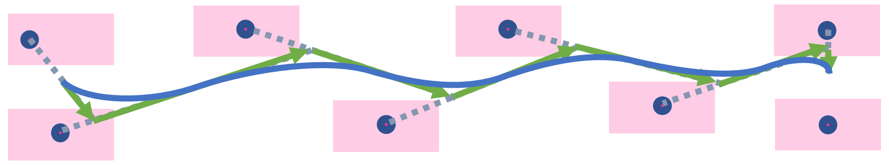
2. Draw the trajectory of the DCM in the backward direction.

*determine the zmp of the first step so that the dcm ends at the initial point



3. Determine the trajectory of the CoM.

*there is no need to compute the CoM trajectory explicitly.



Balance Control

Review of Existing Stabilizers

Name	Description	Measure/Estimate
Torso position compliance control [Nagasaka 1999]	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	

Balance Control

Recall again the following.

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

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

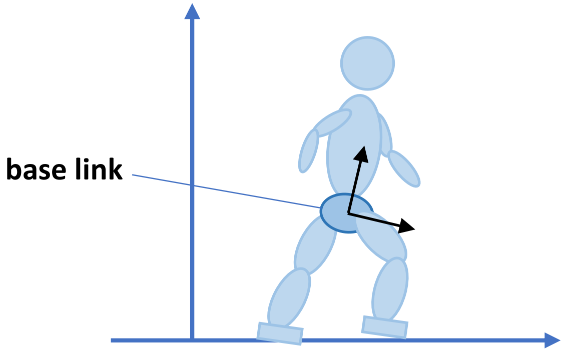
$$\dot{L}_{\mathrm{x}} = \frac{mh}{T}\left(\dot{p}_{\mathrm{dcm,y}} - \frac{1}{T}(p_{\mathrm{dcm,y}} - p_{\mathrm{vrp,y}})\right)$$
$$\dot{L}_{\mathrm{y}} = -\frac{mh}{T}\left(\dot{p}_{\mathrm{dcm,x}} - \frac{1}{T}(p_{\mathrm{dcm,x}} - p_{\mathrm{vrp,x}})\right)$$

Deviation from the DCM dynamics generates change of angular momentum.

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

Balance Control

Let $\theta_{\mathrm{x,y}}$ and $\omega_{\mathrm{x,y}}$ 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 $\theta_{\mathrm{x,y}}$ and $\omega_{\mathrm{x,y}}$ to maintain the orientation of the base link.
To realize this, consider the following feedback law.

Desired moment to recover balance

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

Balance Control

Next, consider the DCM dynamics where the balance recovery moment is input to it 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.

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

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

desired VRP

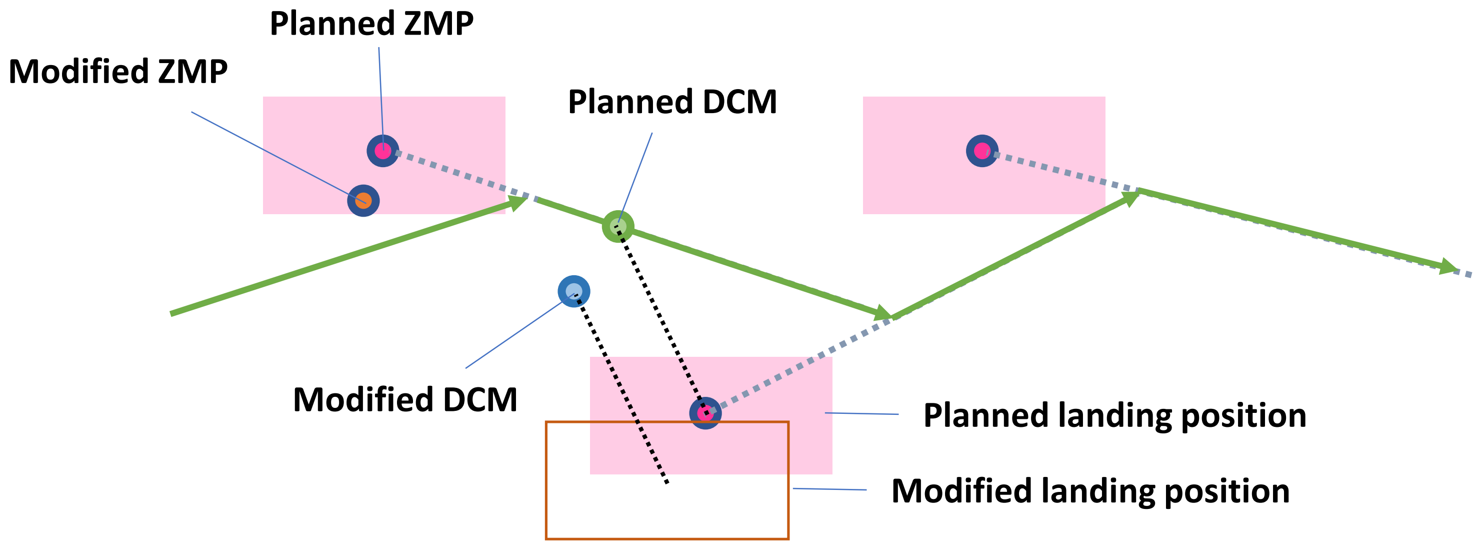
reference VRP

desired DCM

reference DCM

Balance Control

To illustrate:



Landing position is adjusted continuously so that the relative position of the modified DCM and the modified landing position is the same as that of the planned DCM and landing position.

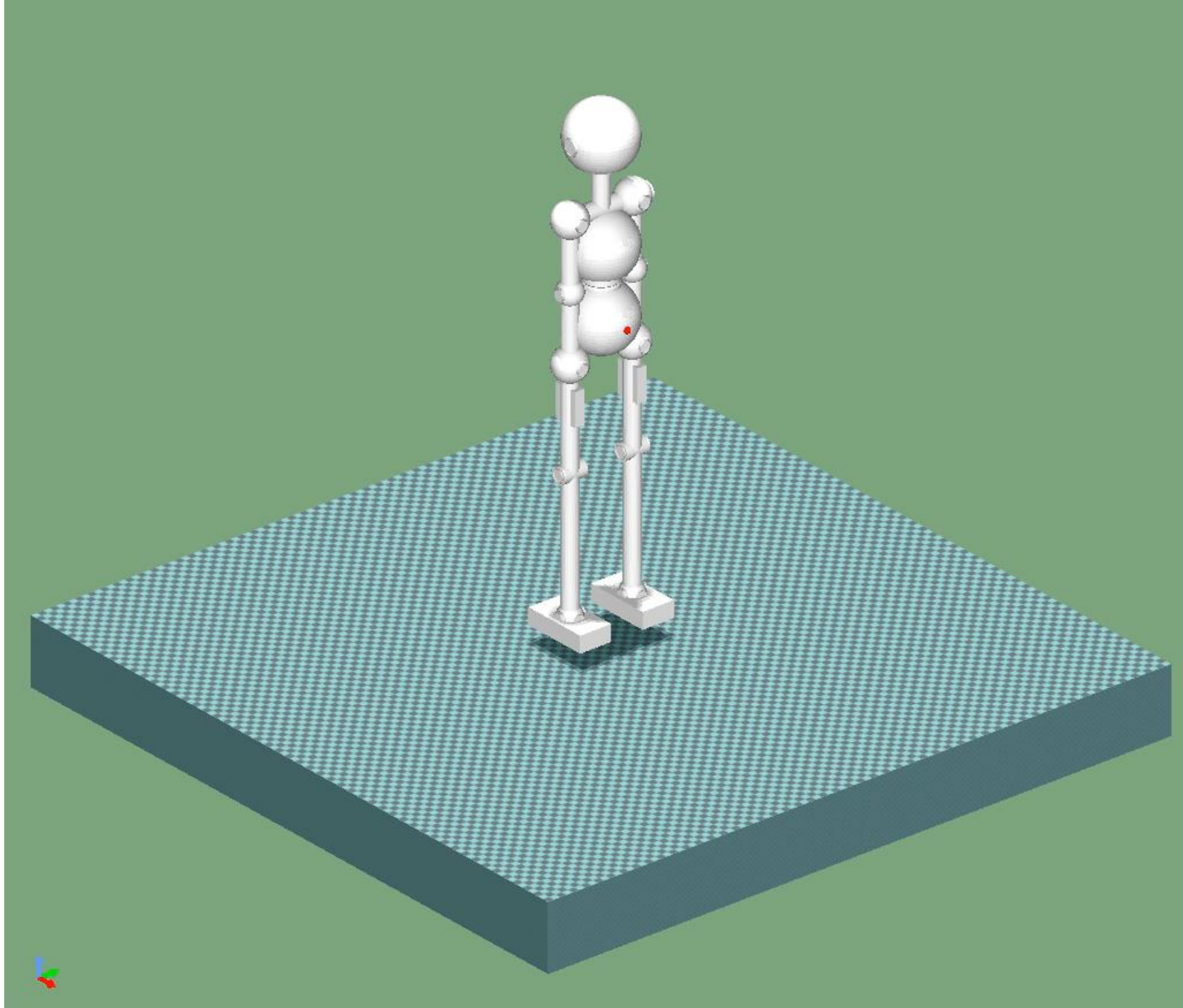
Examples

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

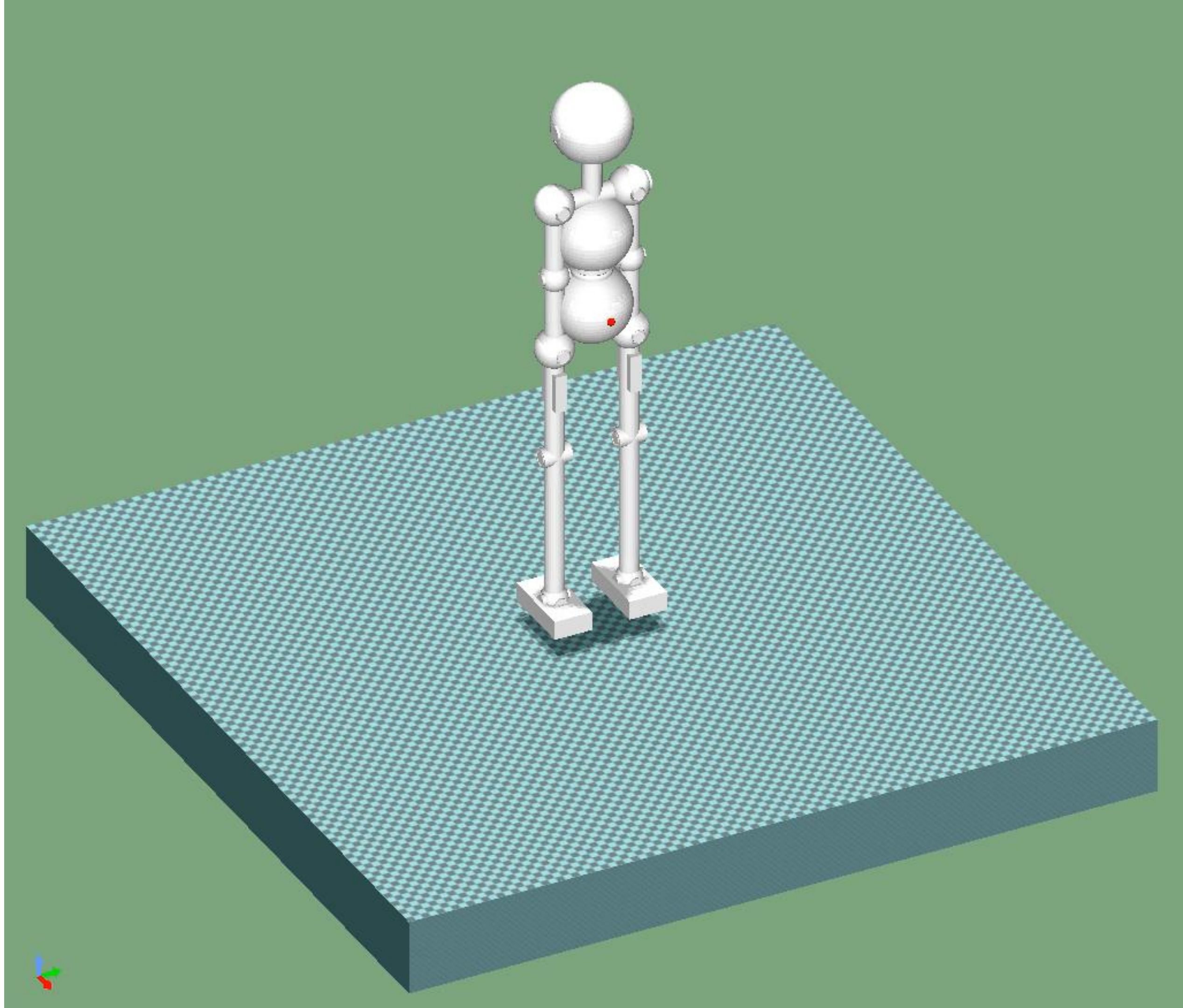


Examples

Simple balance control demo 2

The tilted floor turns with respect to the vertical axis

Tilt angle: 0.1 [rad]
Turning frequency: 2.0 [rad/s]

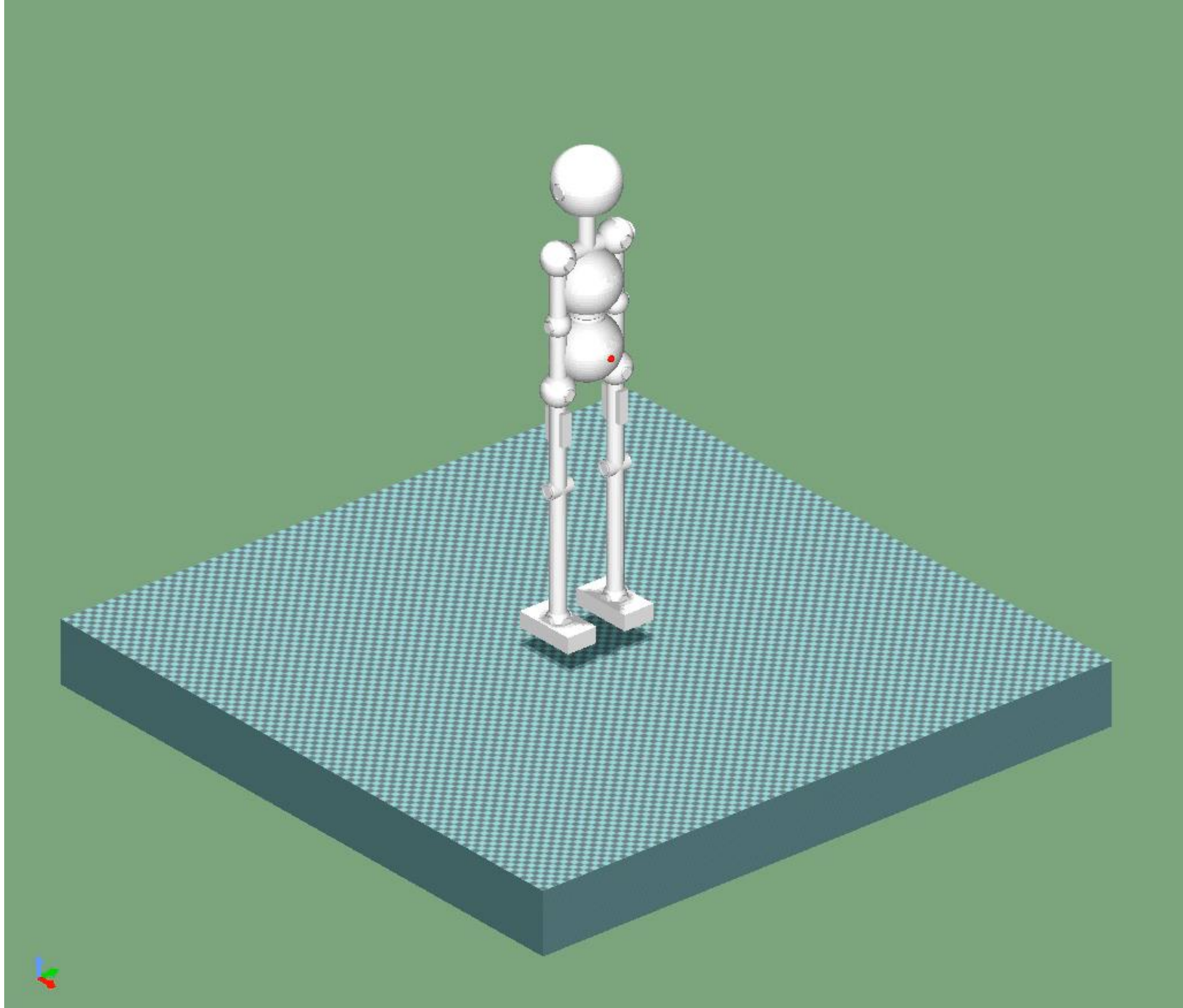


Examples

Simple balance control demo 3

The floor moves horizontally in a sinusoidal pattern

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



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 and flight?

Generate motions without pre-planned contact?

Beyond the scope of this talk

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