1 LIPM

- Continue on path of using simple abstractions
- What simple models of locomotion have you already seen?
  - Rimless wheel
  - Compass gait model
  - SLIP model
- Want something even simpler: linear inverted pendulum model (LIPM)
- Several ways to interpret:
  - **approximation of a simple model**: linearization of inverted pendulum
  - **simple model** in its own right
  - **projection of the full dynamics**, subject to certain constraints.
- Derivation as simple model in its own right.
  - 2D, but will generalize easily to 3D
  - Drawing
  - Similar triangles
  - Newton
  - State space model
  - Phase portrait (compare to pendulum 'eye')
  - Eigenvalues: $\pm\sqrt{\frac{g}{z}}$; one stable, one unstable (saddle point)
  - Taking a step: horizontal translation
- Derivation as projection of full dynamics
  - why? identify exactly which parts of full model correspond to parts of simple model.
  - step back:
    - two point masses
    - equations of motion
    - sum
    - internal forces cancel
    - generalize
  - external forces acting on a humanoid robot: gravity and contact forces
  - same LIPM drawing, but:

    * assume that angular momentum remains constant
    * point mass is CoM
    * rod becomes *any* linkage
    * base is CoP (or ZMP; but note that there are differences in definition)
    * can actually also drop the constraint of no angular momentum (CMP), but there's no notion of 'average angular position', like the CoM
  - Where can the CoP be? No forces exerted at a distance. Base of support

- What do we have now?
  - **linear** model that captures the most important features of walking: how CoM moves in response to CoP
  - **sequence of footsteps** and timing: set of time-varying **linear** constraints on CoP

- What can we do:
  - plan a CoM / CoP trajectory that minimizes some quadratic cost subject to linear constraints: Quadratic Program
  - solve that program **online** at a high rate: **linear** MPC
  - constantly updated stream of CoP's / center of mass accelerations

2 Whole-body control

How do we control the full robot using this plan for the simple model?

- Depends on your robot:
  - position controlled: use inverse kinematics
  - force controlled: more interesting

- Take a step back in complexity: fully actuated robot arm
  - what would you do?
  - Equations of motion (manipulator equations):

$$M\left(q\right)\ddot{q}+c\left(q,\dot{q}\right)=\tau$$

  - Given: desired motions of the joints, expressed as $\ddot{q}_{\text{des}}$, compute τ.
  - Feedback linearization / inverse dynamics / computed torque method.
  - Source of $\ddot{q}_{\text{des}}$: for example, just PD control.

- Can generalize a bit, formulate a QP:

$$
\begin{aligned}
\text{minimize}_{\ddot{q},\tau} \; & \|\tau\|^2 \\
\text{subject to} \; & M(q)\ddot{q} + c(q,\dot{q}) = \tau \\
& D\ddot{q} = e
\end{aligned}
$$

- Point acceleration task:

$$
\begin{aligned}
p &= f(q) \\
\dot{p} &= \frac{\partial f(q)}{\partial q}\dot{q} = J\dot{q} \\
\ddot{p} &= J\ddot{q} + \dot{J}\dot{q} = \ddot{p}_{\text{des}} \\
\underbrace{J}_{D}\ddot{q} &= \underbrace{\ddot{p}_{\text{des}} - \dot{J}\dot{q}}_{e}
\end{aligned}
$$

- CoM task (!)
- Orientation task

- Now: back to the underactuated case.
  - Equations of motion:

$$M(q)\ddot{q} + c(q,\dot{q}) = Bu + J_c^T(q)\lambda$$

  - Forces:

$$\lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_2 \end{pmatrix}, \quad \lambda_i \in \text{friction cone}$$

  - Updated optimization problem:

$$
\begin{aligned}
\text{minimize}_{\ddot{q},u,\lambda} \; & \|\tau\|^2 \\
\text{subject to} \; & M(q)\ddot{q} + c(q,\dot{q}) = Bu + J_c^T(q)\lambda \\
& D\ddot{q} = e \\
& \lambda \in \text{friction cones} \\
& u_{\min} \le u \le u_{\max}
\end{aligned}
$$

  - Approximate using QP
    * $\lambda_1 = \sum_i \rho_i \beta_i,\ \rho_i \ge 0$
    * run at 500 - 1000 Hz