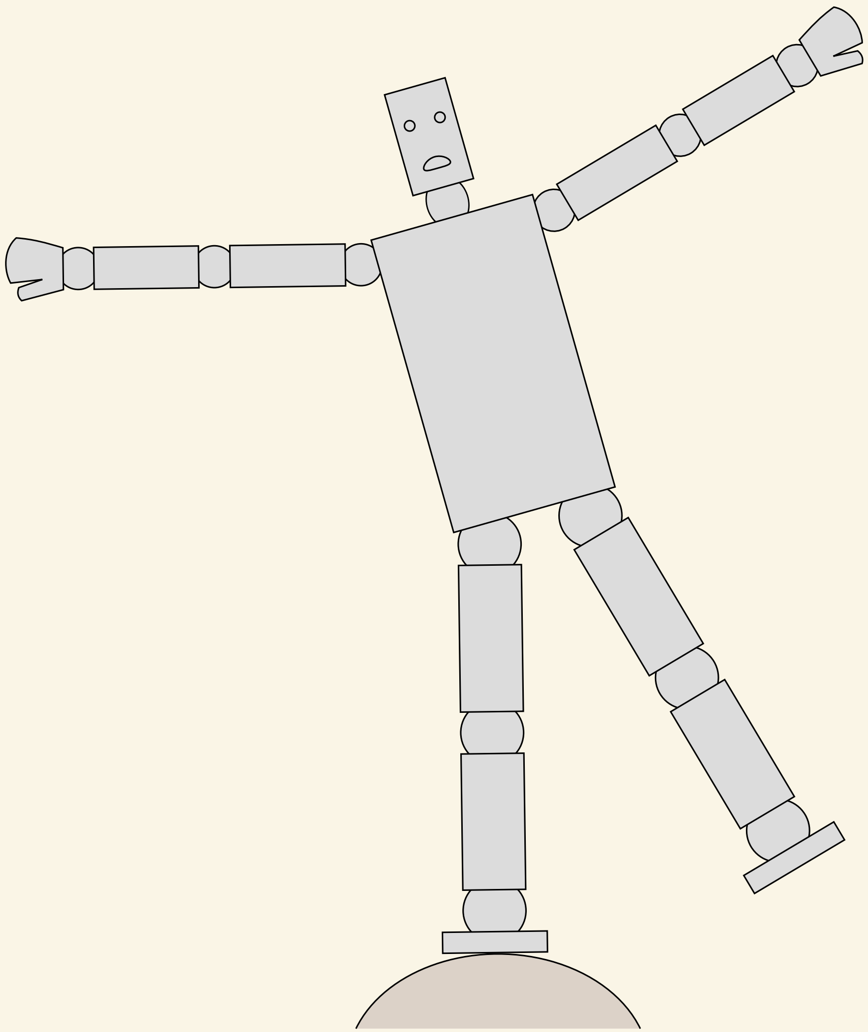
The Physics and Control of Balancing on a Point

Roy Featherstone

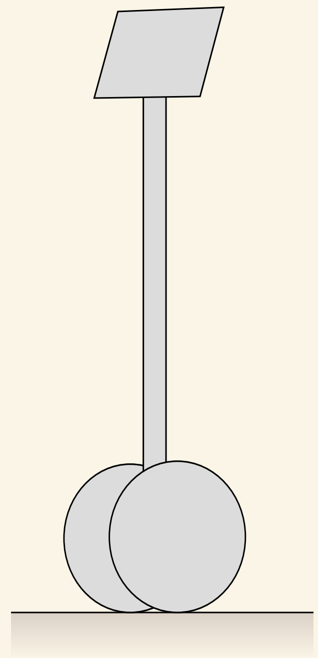
2015

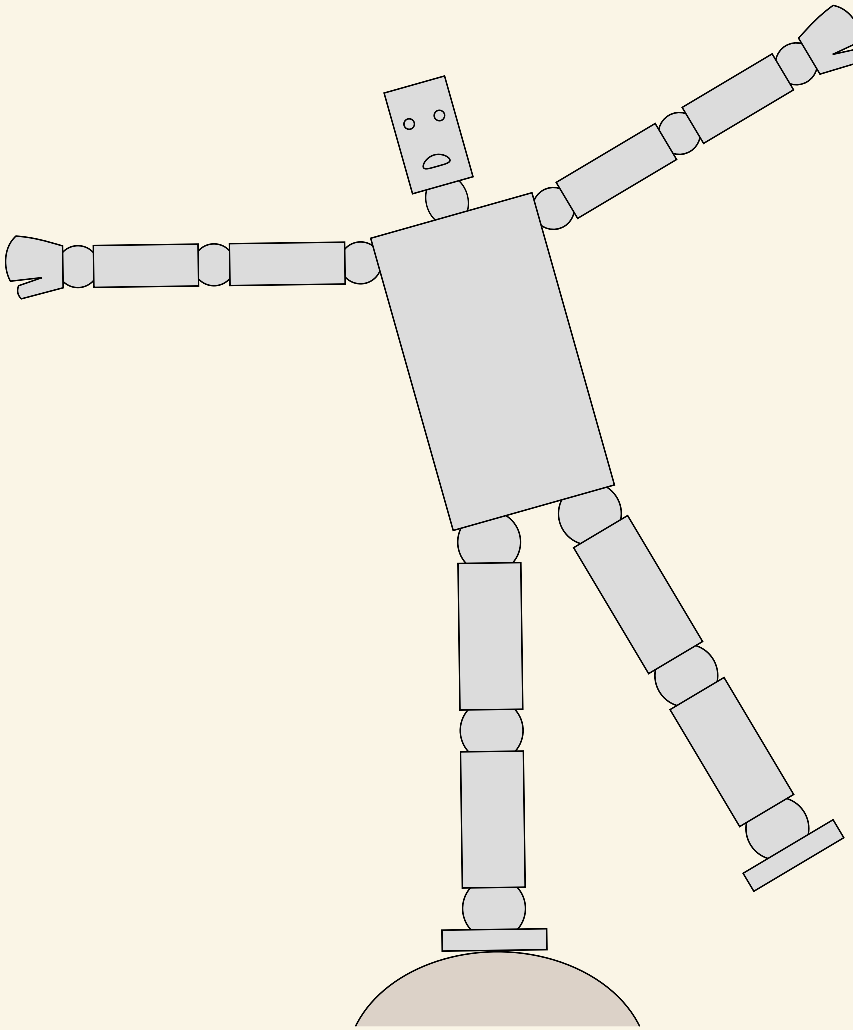
iit ADVR
ISTITUTO ITALIANO DI TECNOLOGIA
ADVANCED ROBOTICS



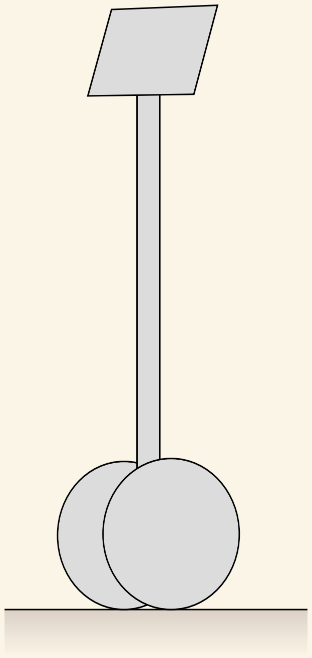
Robots do not always have a polygon of support.

Sometimes they have to balance actively.

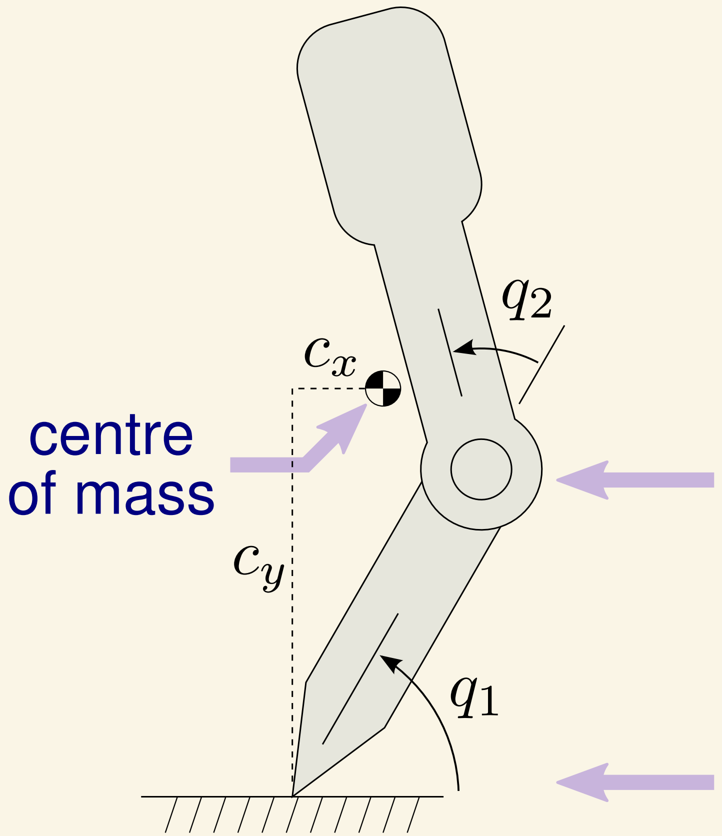




Balancing is usually seen as a control problem, but it is also a *physical process*, and can be analysed as such.



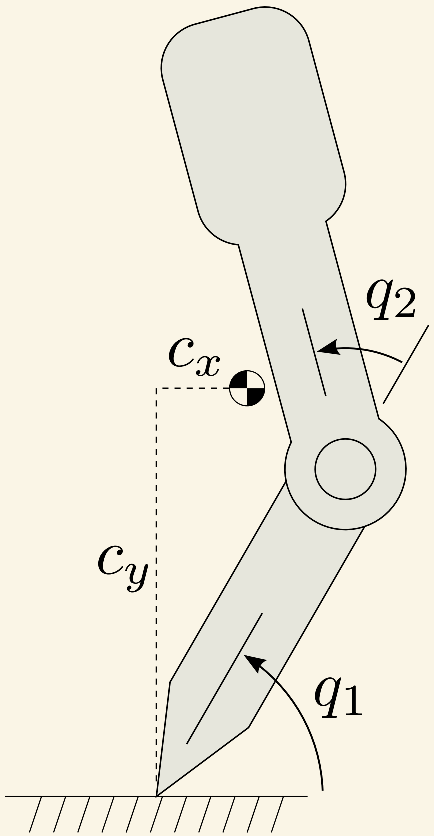
Physics of Balancing on a Point



The simplest case:

A planar double pendulum with an actuated joint balancing on a sharp point in 2D (a knife edge in 3D).

Physics of Balancing on a Point



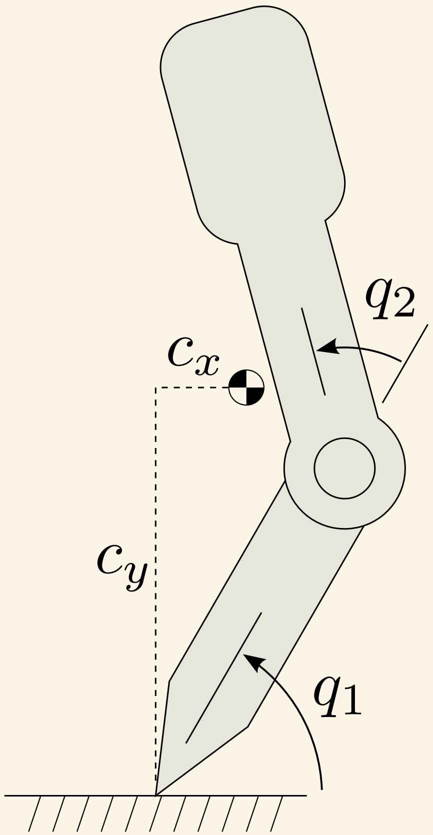
Objectives:

1. Maintain balance: $c_x = \dot{c}_x = 0$
2. Follow commanded motion: $q_2 = q_{2c}$ $\dot{q}_2 = \dot{q}_{2c}$

The control problem:

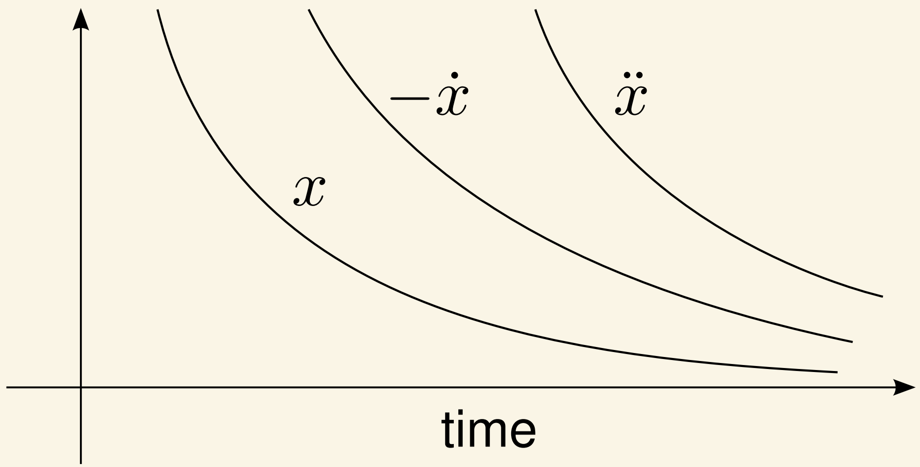
The controller must control 4 variables (c_x, $\dot{c}_x$, q_2 and $\dot{q}_2$), but has direct control of only one variable: τ_2

Physics of Balancing on a Point

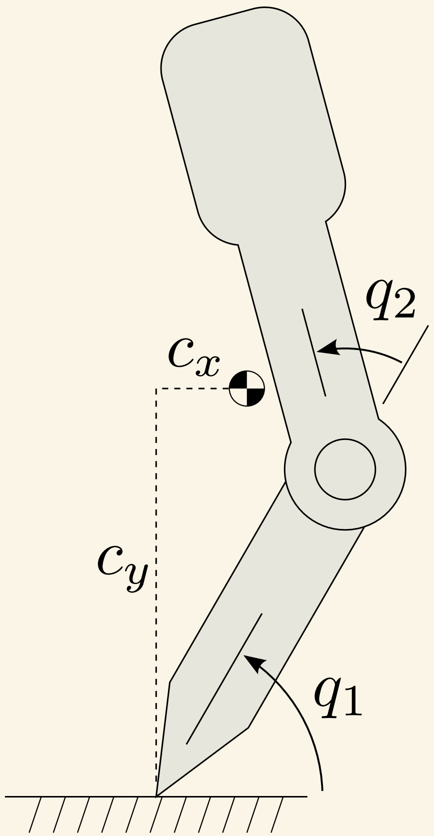


The control solution: (in principle)

If a control system succeeds in driving a variable x to zero, then a side-effect is to drive $\dot{x}$, $\ddot{x}$, etc. also to zero.



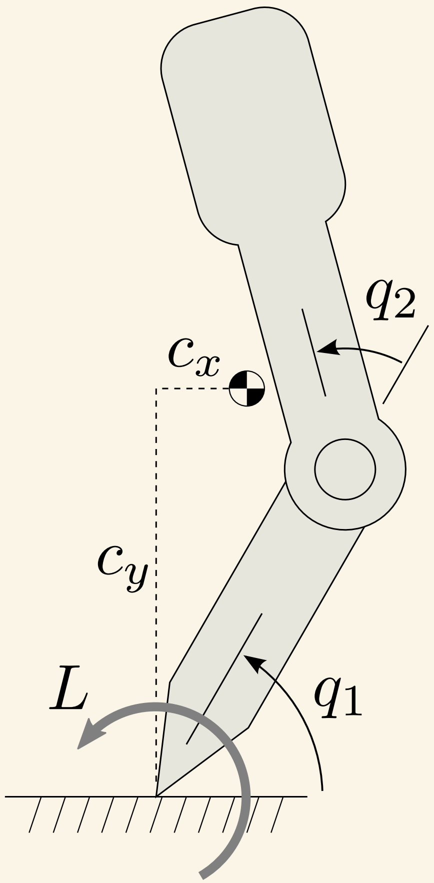
Physics of Balancing on a Point



The control solution: (in principle)

So we seek a new set of state variables to use in place of q_1, q_2, $\dot{q}_1$ and $\dot{q}_2$ with the property that controlling one has the side-effect of controlling the other three.

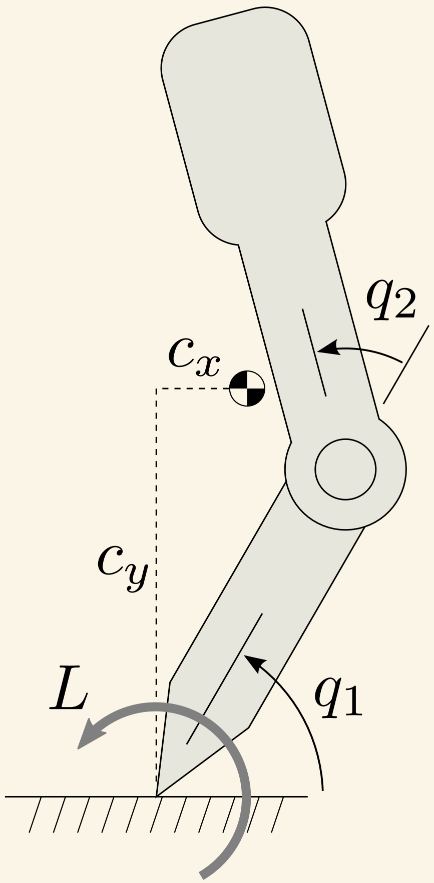
Physics of Balancing on a Point



Analysis:

Let L be the angular momentum of the robot about the support point. L has the special property that $\dot{L}$ is the moment of gravity about the support point.

Physics of Balancing on a Point



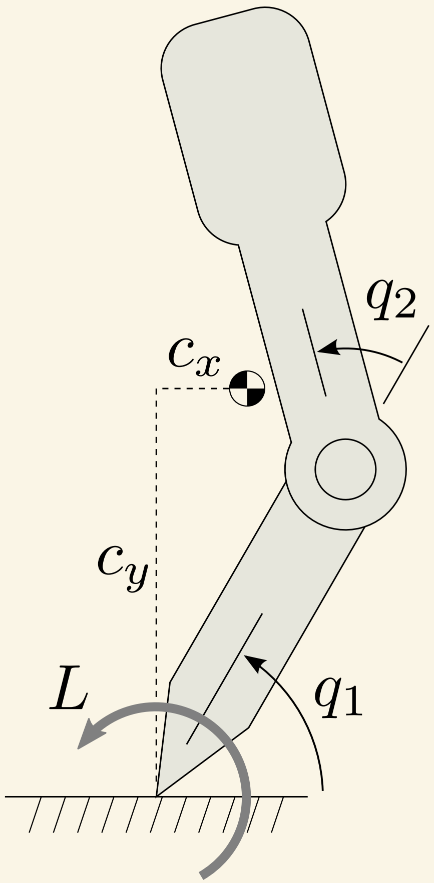
Analysis:

$$L = H_{11}\dot{q}_1 + H_{12}\dot{q}_2$$
$$\dot{L} = -mgc_x$$
$$\ddot{L} = -mg\dot{c}_x$$

Where H_{ij} are elements of the joint-space inertia matrix, m is the mass of the robot, and g is the acceleration of gravity.

Observe that L and $\ddot{L}$ are *linear* functions of velocity.

Physics of Balancing on a Point



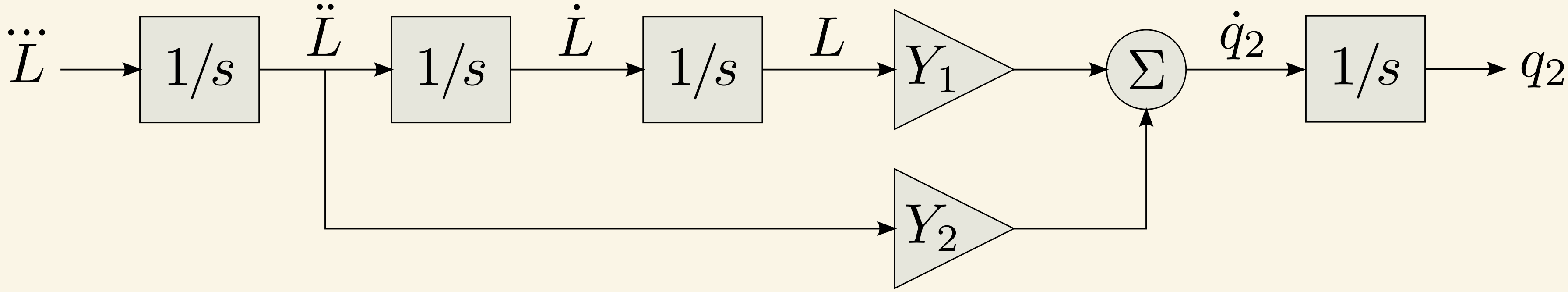
Analysis:

As L and $\ddot{L}$ are linear functions of $\dot{q}_1$ and $\dot{q}_2$, we can invert the equations and write

$$\dot{q}_2 = Y_1 L + Y_2 \ddot{L}$$

where Y_1 and Y_2 are functions of q_1 and q_2 only, and can be calculated easily via standard dynamics algorithms.

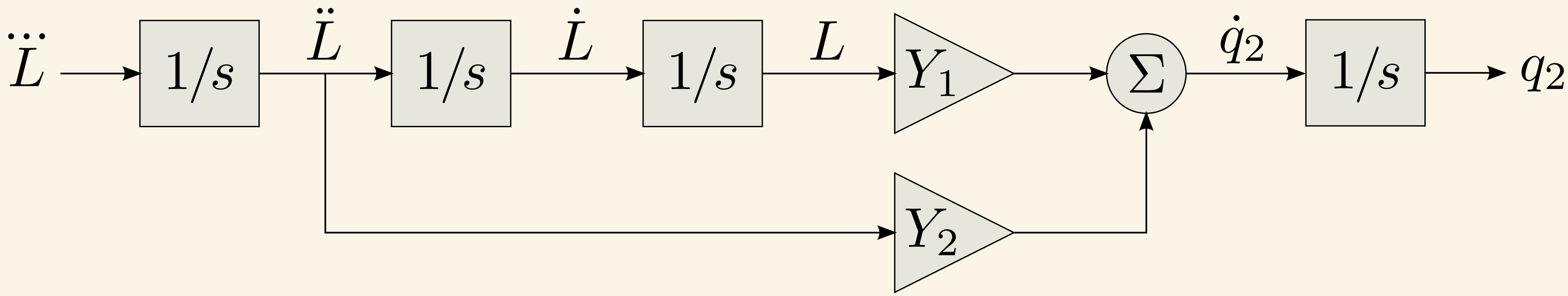
New Model of Balancing



The result is a new model of the balancing behaviour of the robot in which

- the state variables are $\ddot{L}$, $\dot{L}$, L and q_2,
- the input is $\dddot{L}$ and the output is q_2,
- controlling q_2 has the side-effect of maintaining the robot's balance

New Model of Balancing

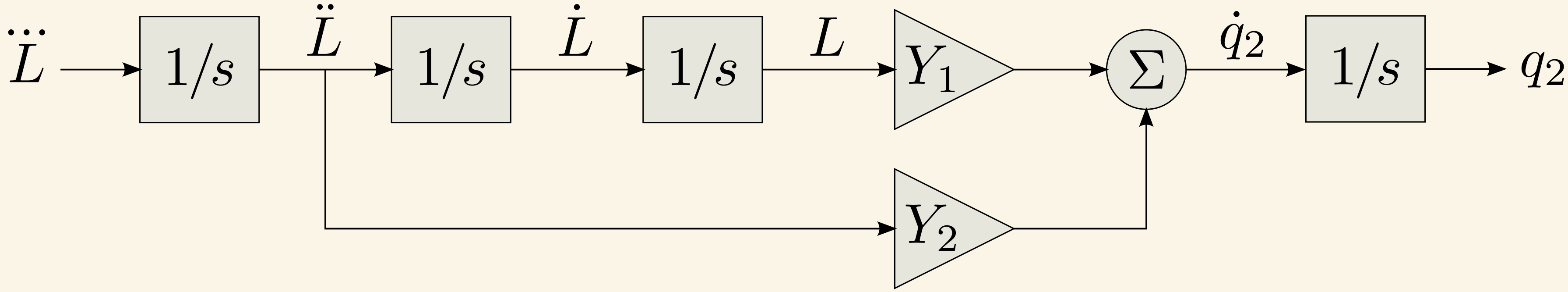


The result is a new model of the bala
in which

$$
\begin{array}{rcl}
q_2 = \text{const} & \Rightarrow & \dot{q}_2 = 0 \\
\dot{q}_2 = 0 & \Rightarrow & L = \dot{L} = \ddot{L} = 0 \\
\dot{L} = 0 & \Rightarrow & c_x = 0 \\
\ddot{L} = 0 & \Rightarrow & \dot{c}_x = 0
\end{array}
$$

- the state variables are $\ddot{L}$, $\dot{L}$
- the input is $\dddot{L}$ and the output is q_2,
- controlling q_2 has the side-effect of maintaining the robot's balance

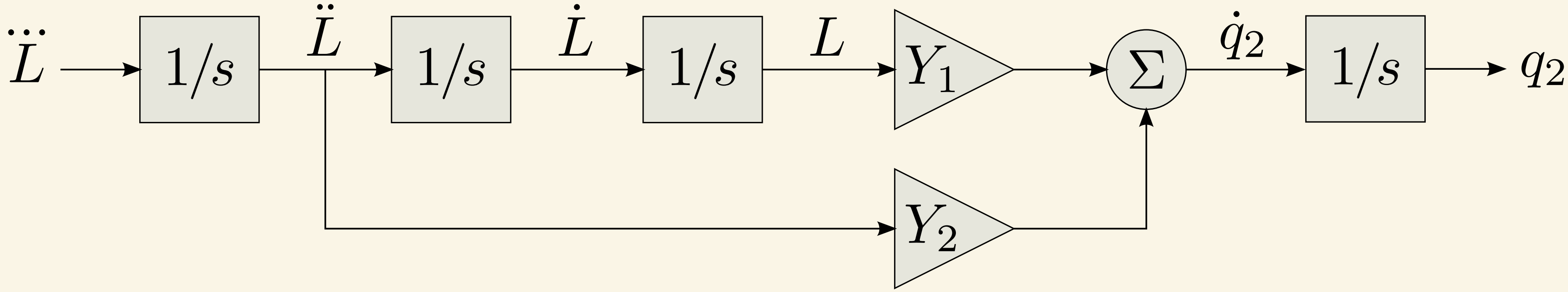
New Model of Balancing



To control the robot we

1. map q_1, $\dot{q}_1$, q_2 and $\dot{q}_2$ to $\ddot{L}$, $\dot{L}$, L and q_2,
2. apply a *simple control law* to calculate $\dddot{L}$,
3. convert $\dddot{L}$ to τ_2 or $\ddot{q}_2$ as required

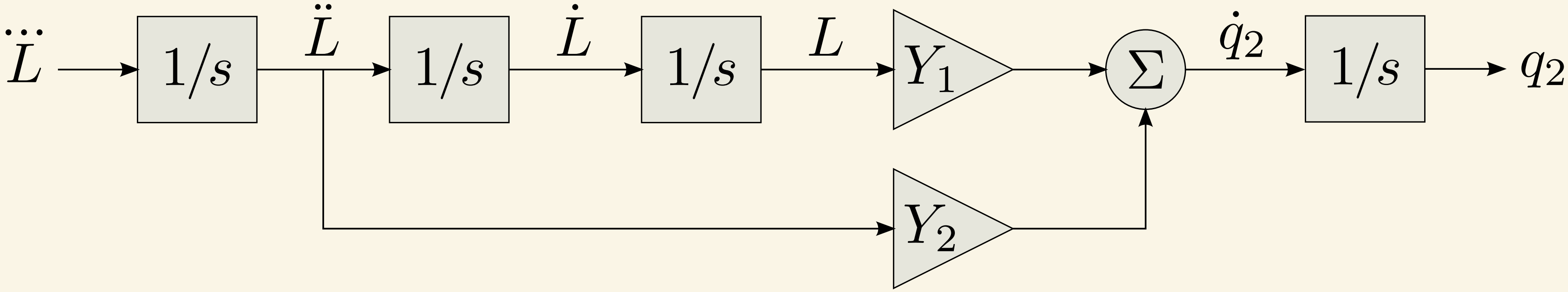
Balance Controller



$$\dddot{L} = k_{dd}(\ddot{L} - \ddot{L}_{\mathrm{c}}) + k_d(\dot{L} - \dot{L}_{\mathrm{c}}) + k_L(L - L_{\mathrm{c}}) + k_q(q_2 - q_{2\mathrm{c}})$$

the gains are simple
functions of Y_1, Y_2 and
the user's choice of poles

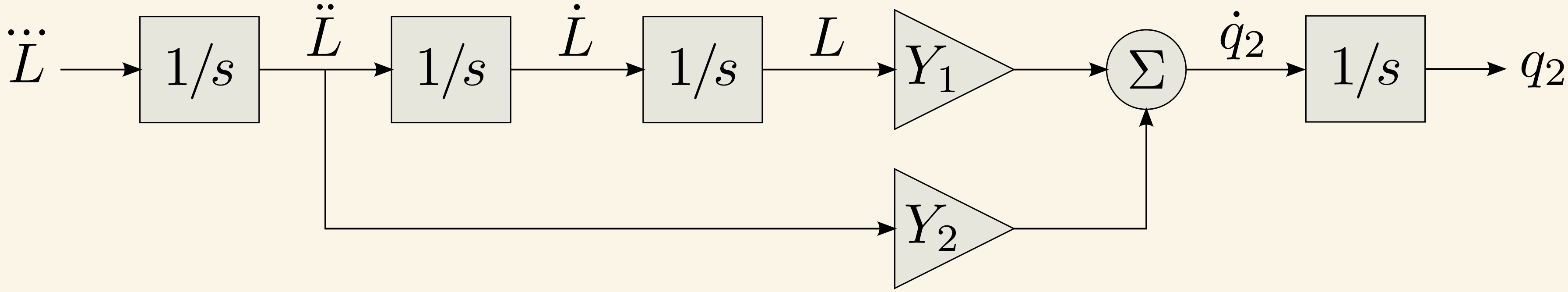
Balance Controller



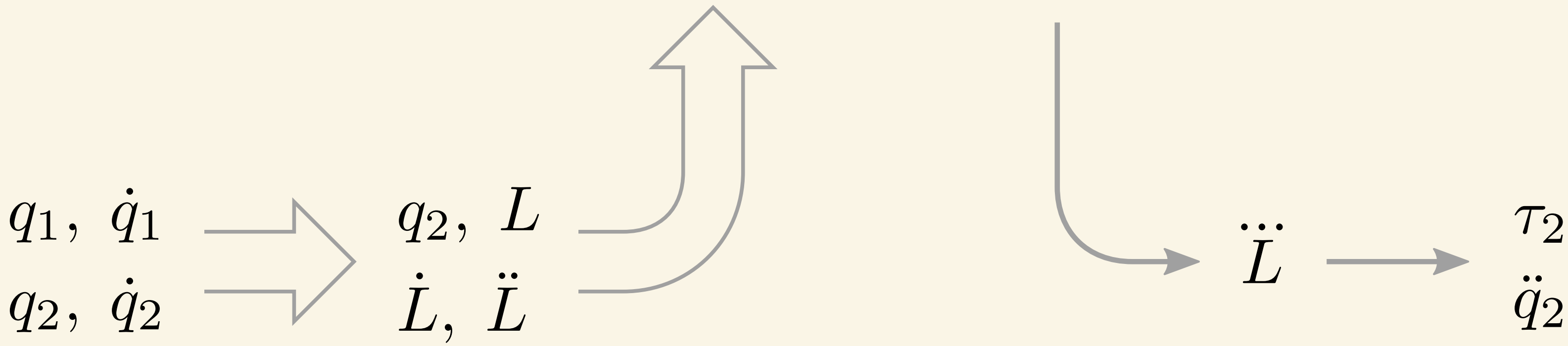
$$\dddot{L} = k_{dd}(\ddot{L} - \ddot{L}_{\mathrm{c}}) + k_d(\dot{L} - \dot{L}_{\mathrm{c}}) + k_L(L - L_{\mathrm{c}}) + k_q(q_2 - q_{2\mathrm{c}})$$

optional

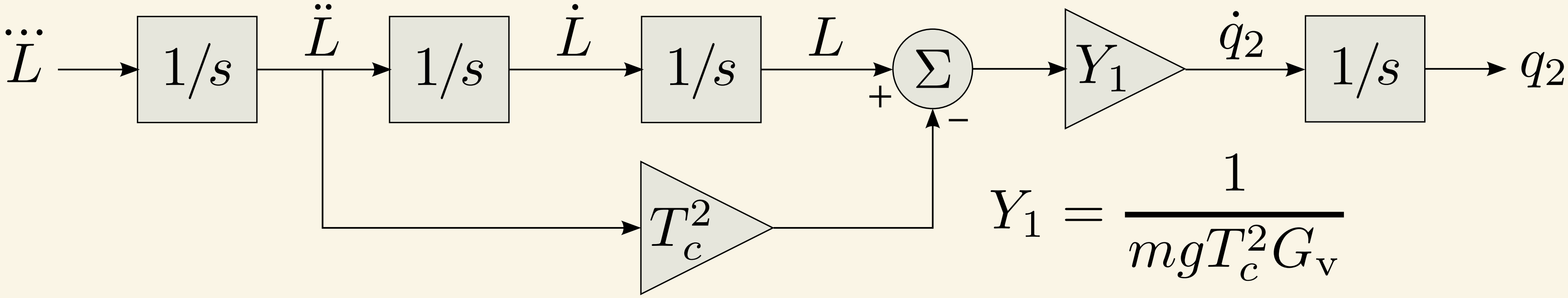
Balance Controller



$$\dddot{L} = k_{dd}(\ddot{L} - \ddot{L}_{\mathrm{c}}) + k_d(\dot{L} - \dot{L}_{\mathrm{c}}) + k_L(L - L_{\mathrm{c}}) + k_q(q_2 - q_{2\mathrm{c}})$$



A Bit More Physics



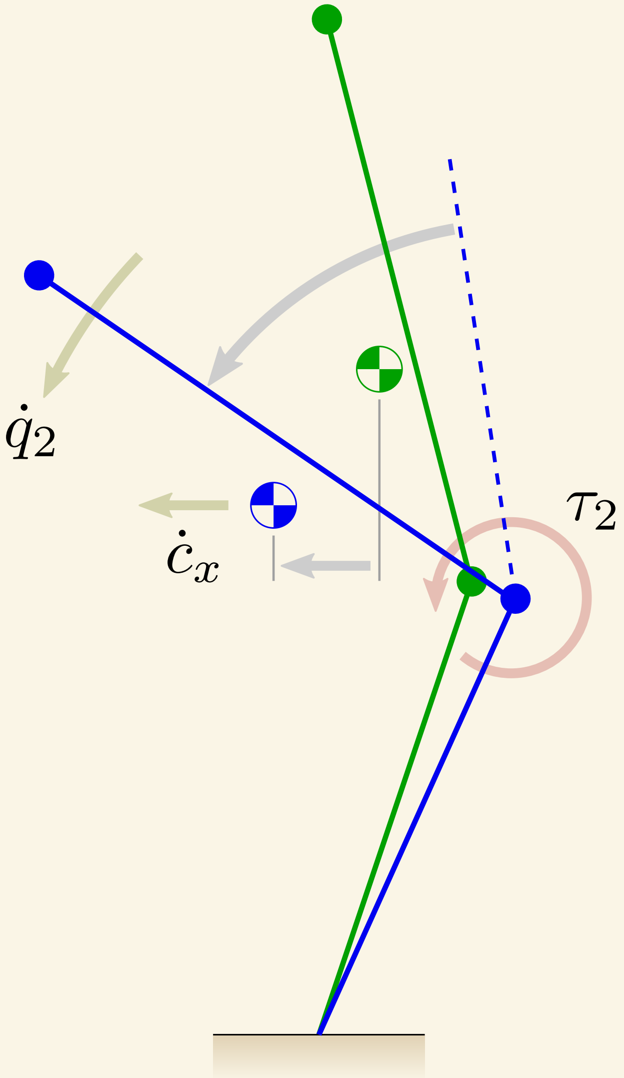
$$Y_1 = \frac{1}{mgT_c^2 G_\mathrm{v}}$$

$$T_c^2 = \frac{-Y_2}{Y_1}$$

where

- T_c is the robot's *natural time constant of toppling*, treating it as a single rigid body
- G_v is the *linear velocity gain* of the robot, which measures the degree to which motion of the actuated joint influences the horizontal motion of the CoM

A Bit More Physics



A robot's *velocity gain* expresses the instantaneous relationship between motion of the actuated joint(s) and the resulting motion of the centre of mass.

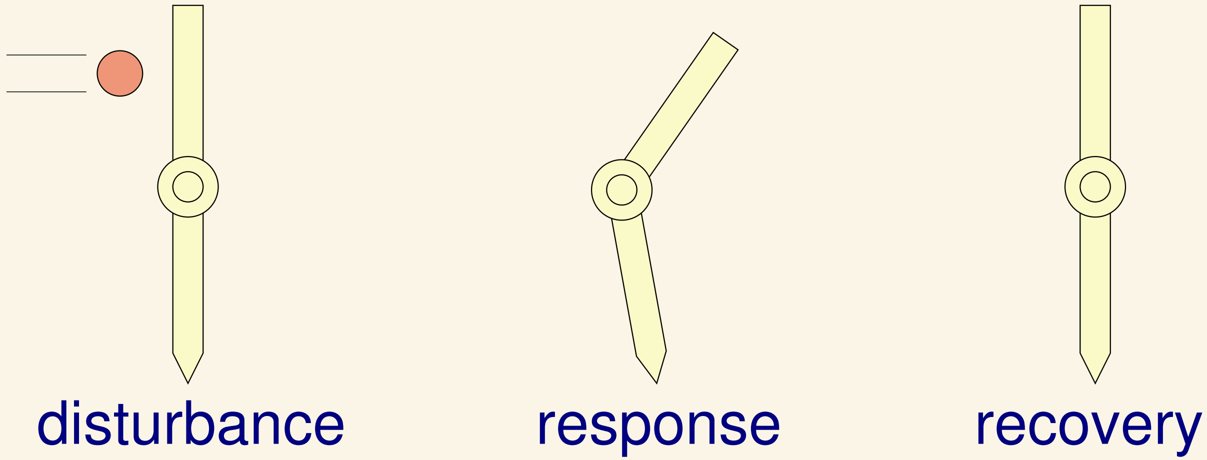
For the double pendulum,

$$G_{\mathrm{v}} = \frac{\Delta \dot{c}_x}{\Delta \dot{q}_2}$$

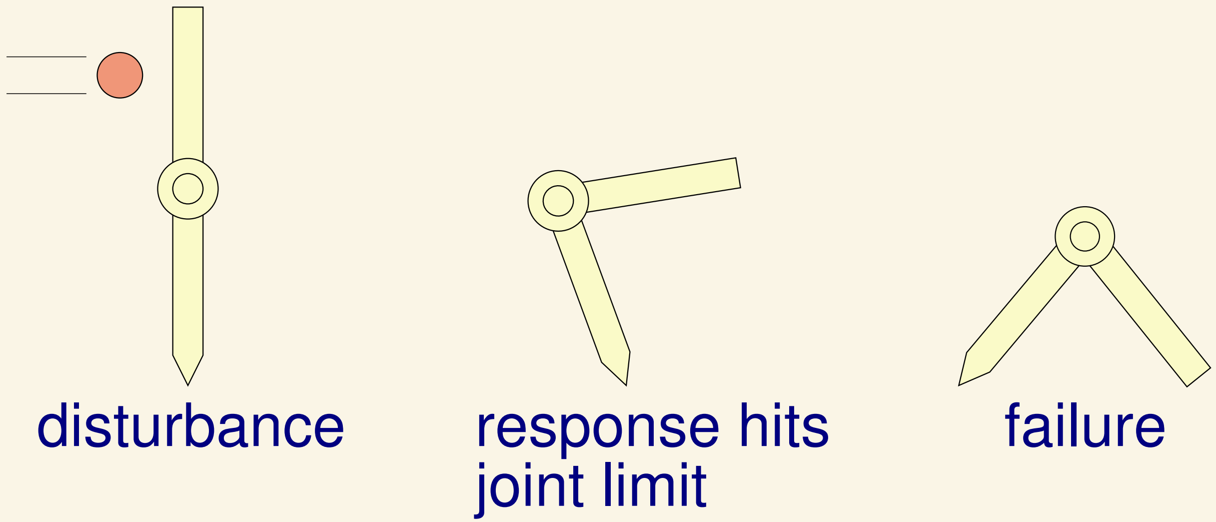
where both velocity changes are caused by an impulse at joint 2.

A Bit More Physics

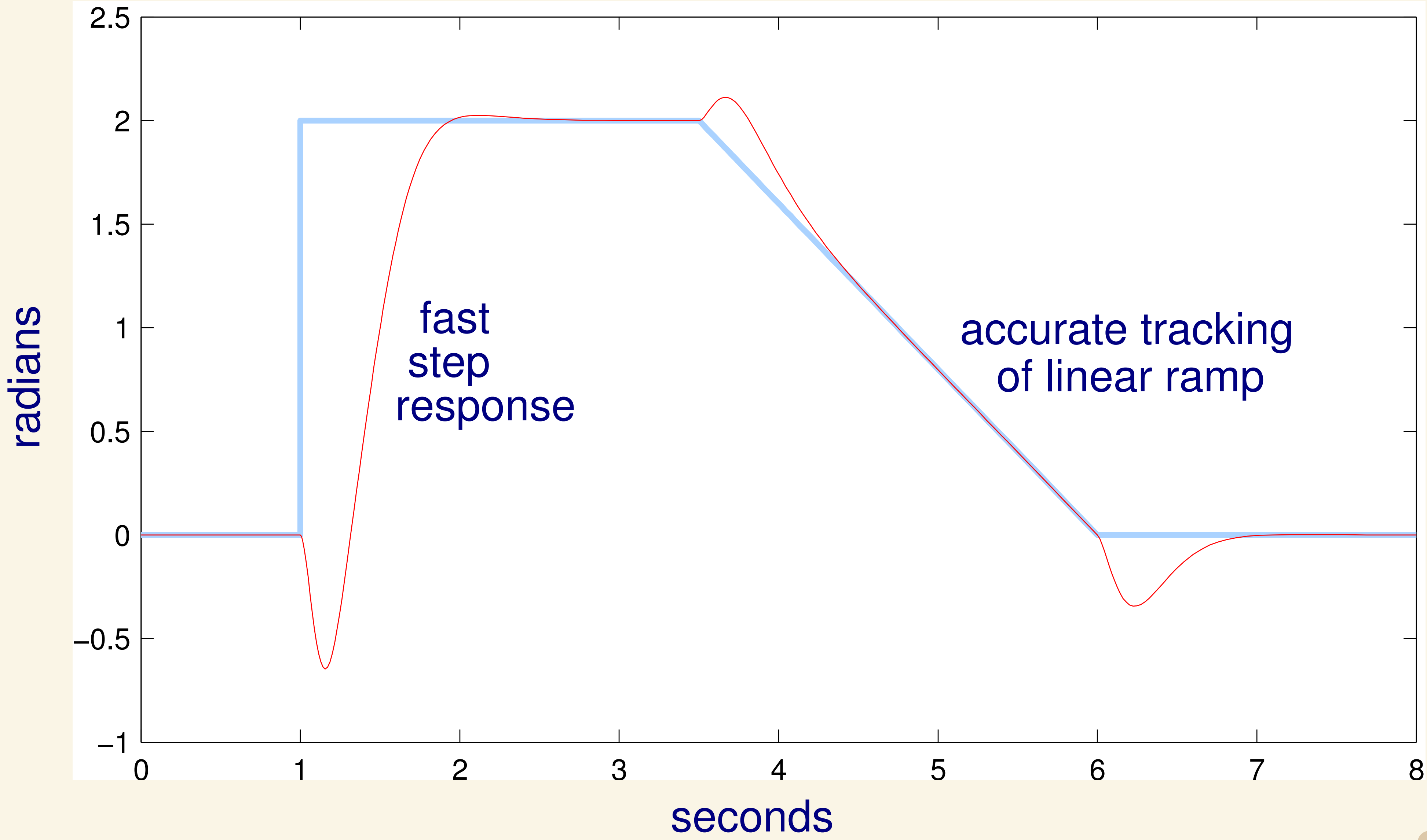
If $|G_{\mathrm{v}}|$ is large then the robot is good at balancing



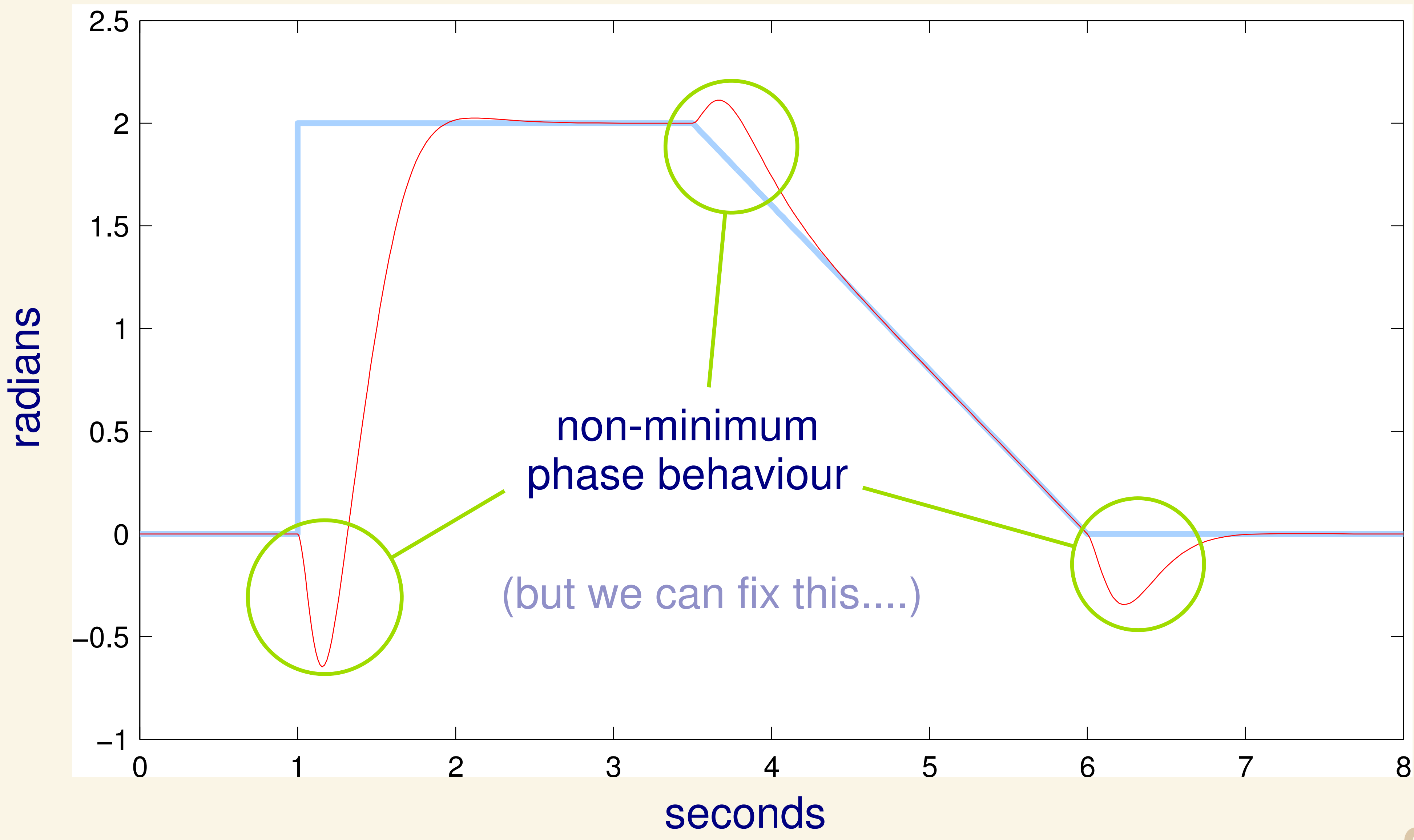
But if $|G_{\mathrm{v}}|$ is small....



How Well Does it Work?

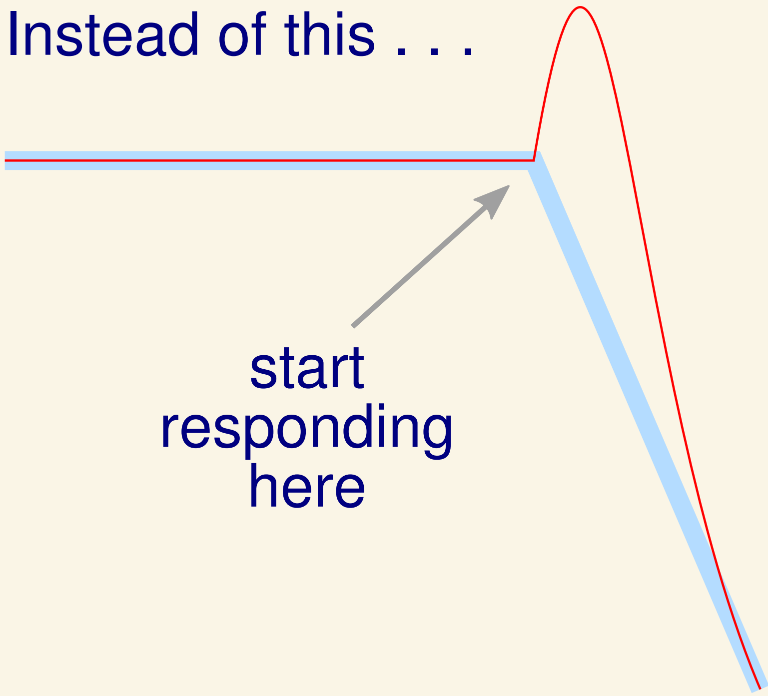


How Well Does it Work?

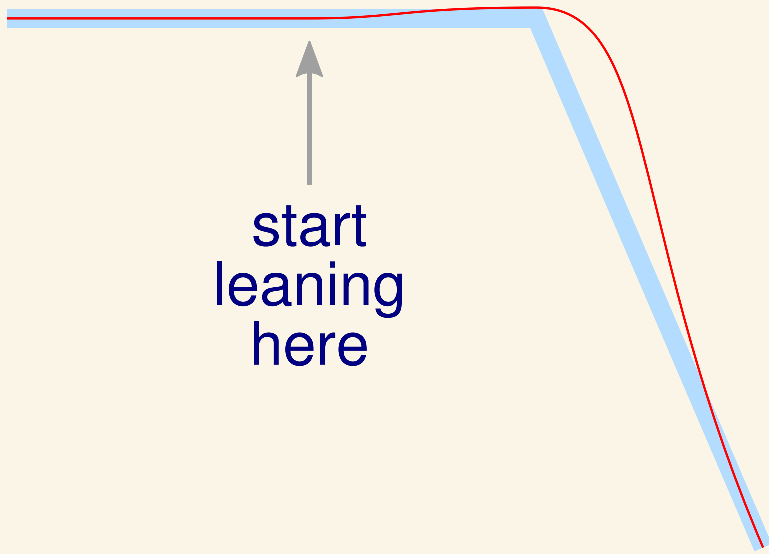


Leaning in Anticipation

Instead of this . . .

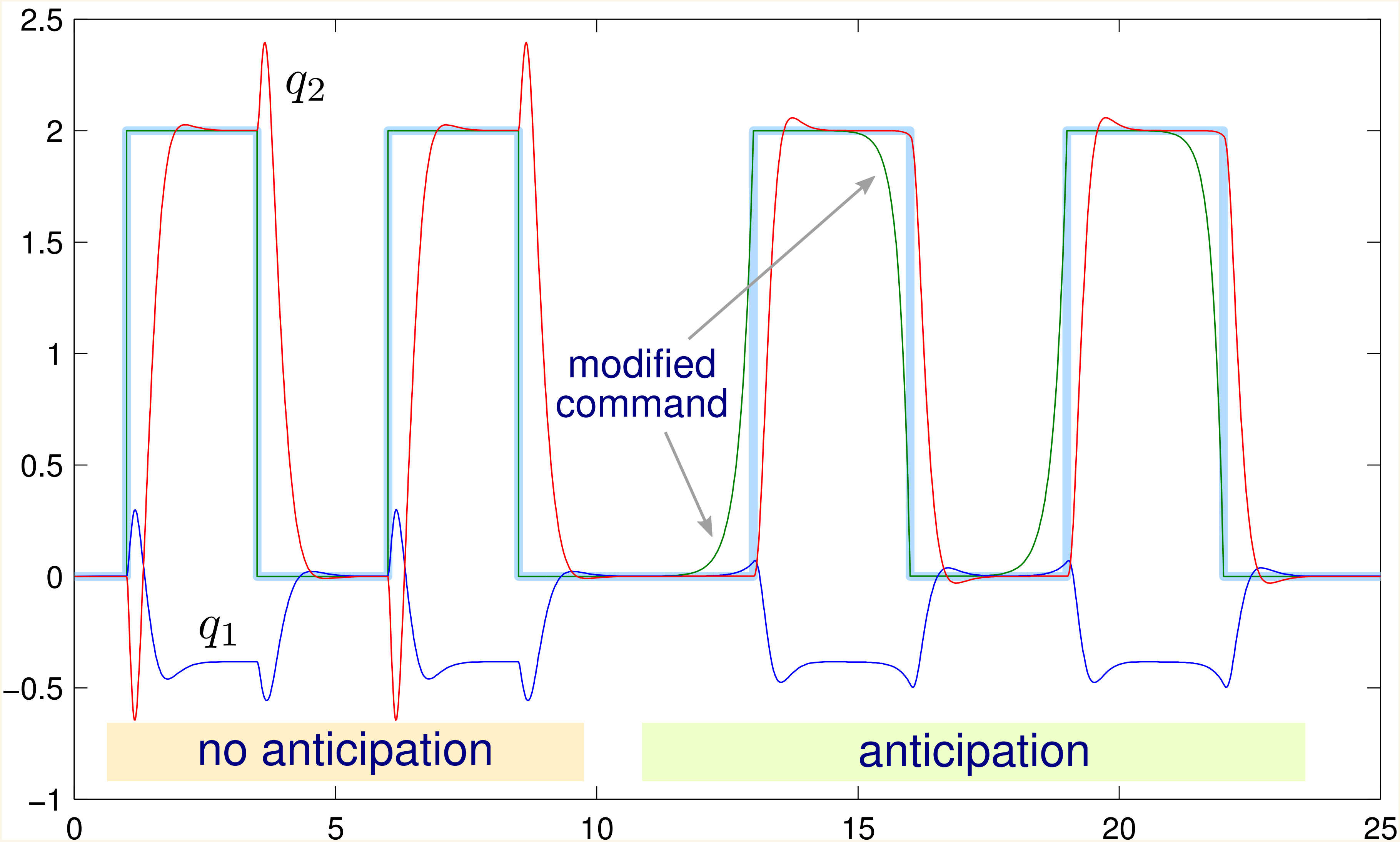


why not do this?



This behaviour can be implemented by changing the command input to the controller.

Leaning in Anticipation



The End

Further reading:

`http://royfeatherstone.org/skippy/`

`http://royfeatherstone.org/publications.html`