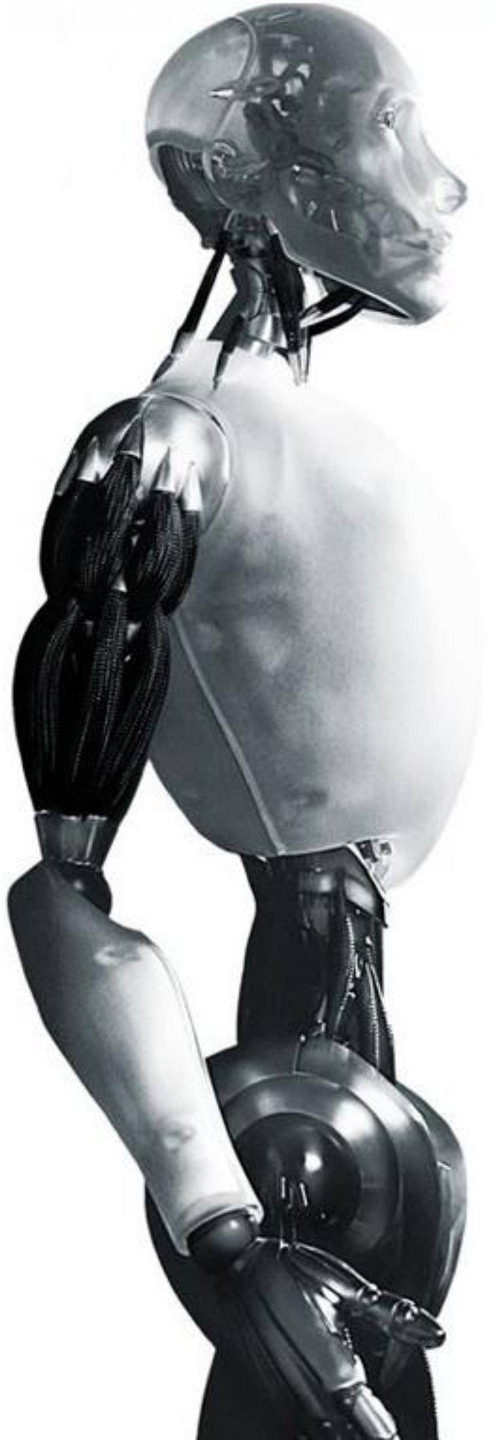
Disclaimer

These slides are intended as presentation aids for the lecture. They contain information that would otherwise be to difficult or time-consuming to reproduce on the board. But they are incomplete, not self-explanatory, and are not always used in the order they appear in this presentation. As a result, these slides should not be used as a script for this course. I recommend you take notes during class, maybe on the slides themselves. It has been shown that taking notes improves learning success.

Reading for this set of slides

- Craig – Intro to Robotics (3rd Edition)
  - Chapter 1

Please note that this set of slides is intended as support for the lecture, not as a
stand-alone script. If you want to study for this course, please use these slides
in conjunction with the indicated chapters in the text books. The textbooks are
available online or in the TUB library (many copies that can be checked out for the
entire semester. There are also some aspects of the lectures that will not be
covered in the text books but can still be part of the homework or exam. For those
It is important that you attend class or ask somebody about what was covered in class.

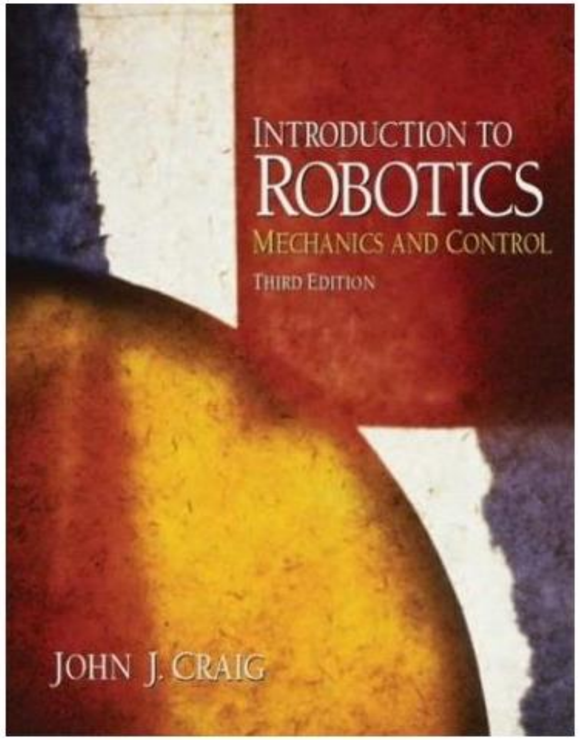


Robotics

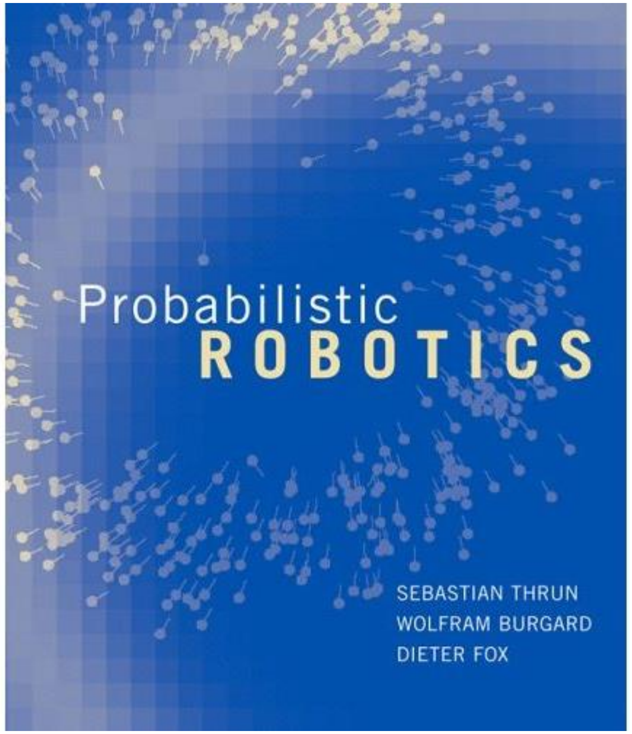
Our First Task: Hold Still!

TU Berlin

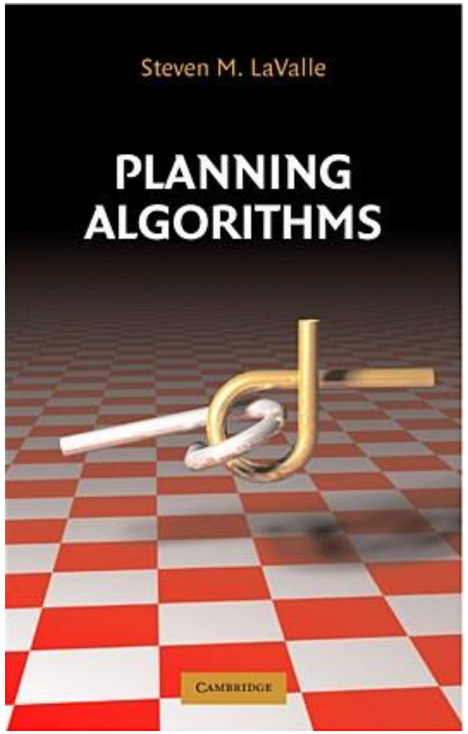
Oliver Brock



In the library



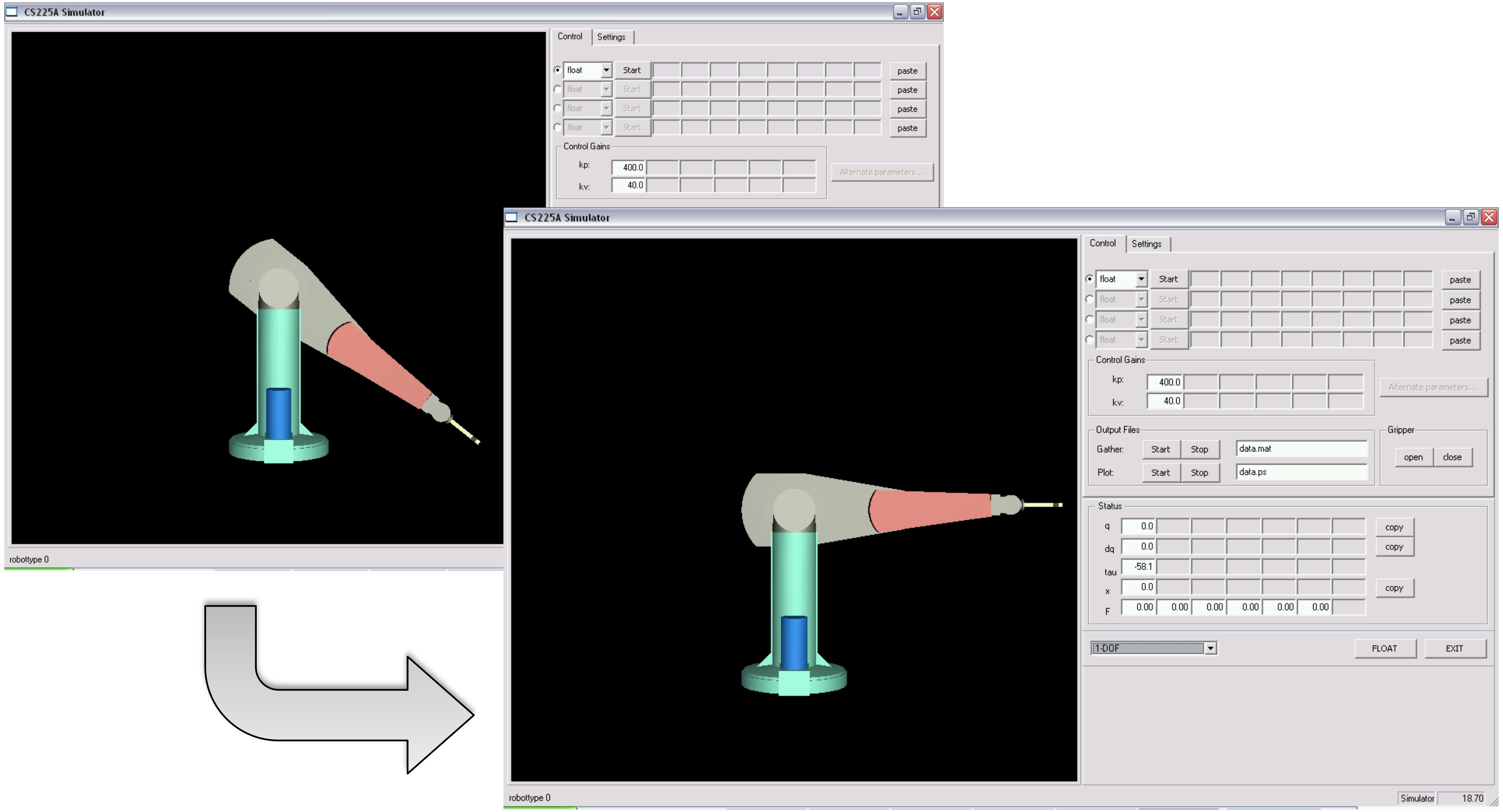
In the library
http://www.probabilistic-robotics.org



Free online
http://planning.cs.uiuc.edu


PUMA 500
UNIMATE

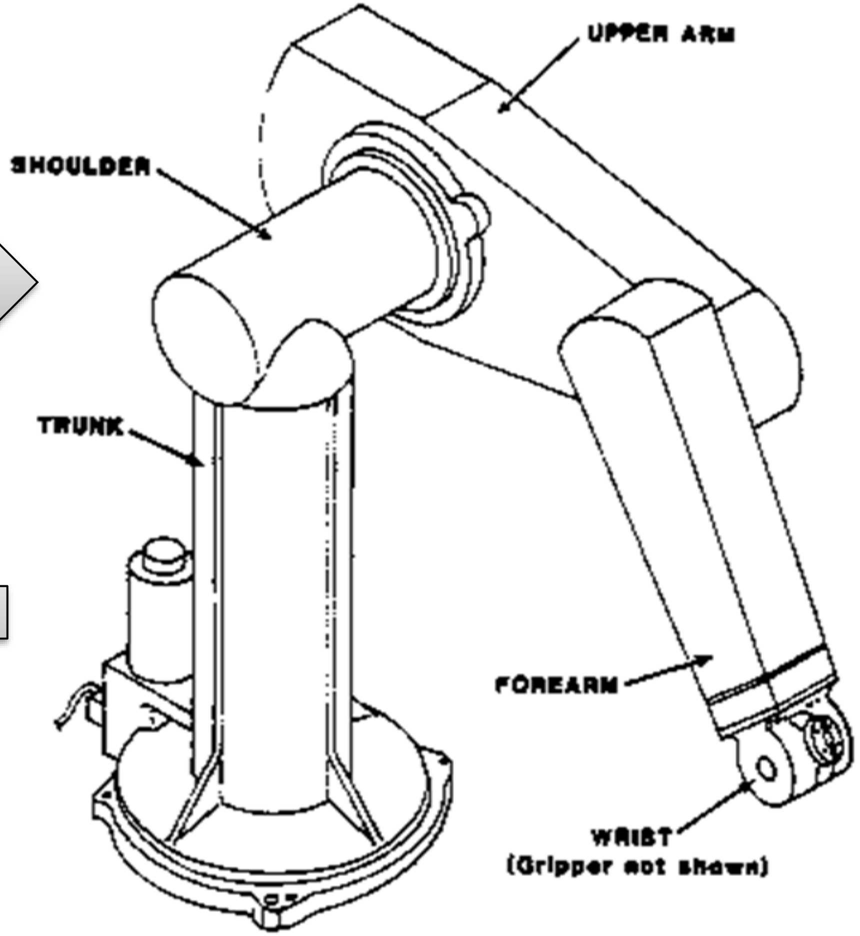
Our Task: Hold Still!



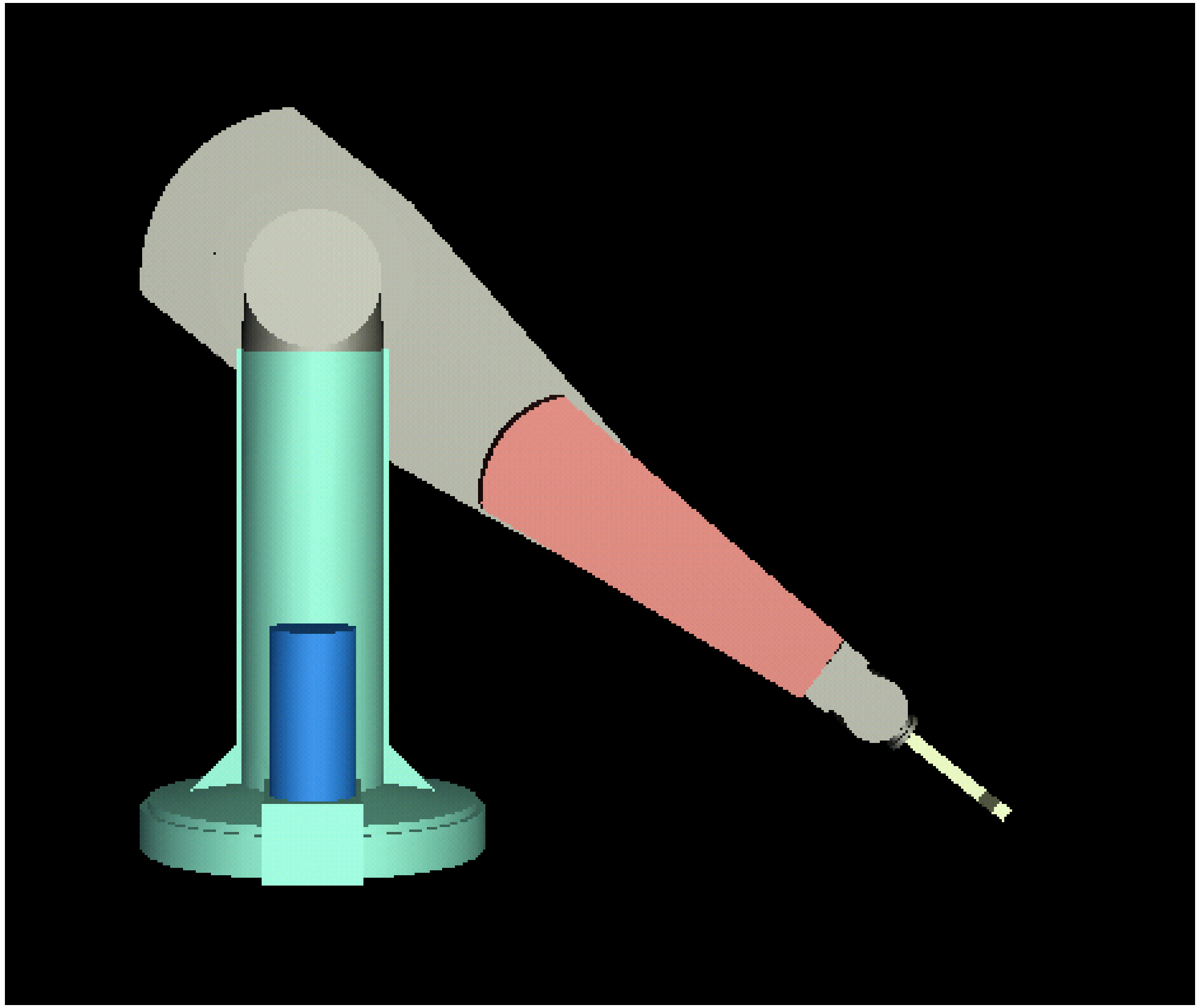
The Interface

torque / force τ (tau)
(really: current)
[Drehmoment, Kraft]

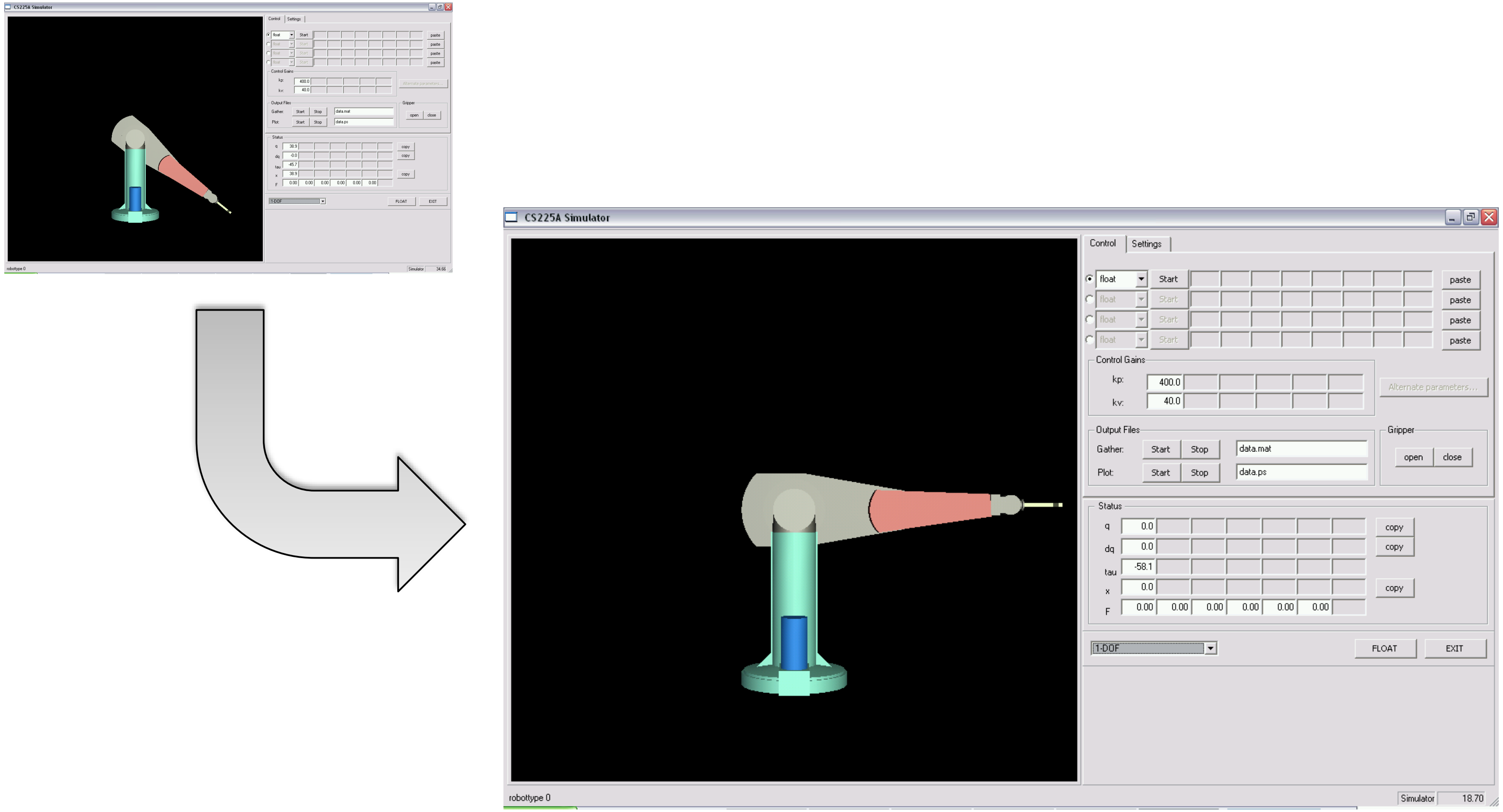
joint positions:
configuration *q*



Don’t fall. Hold still!

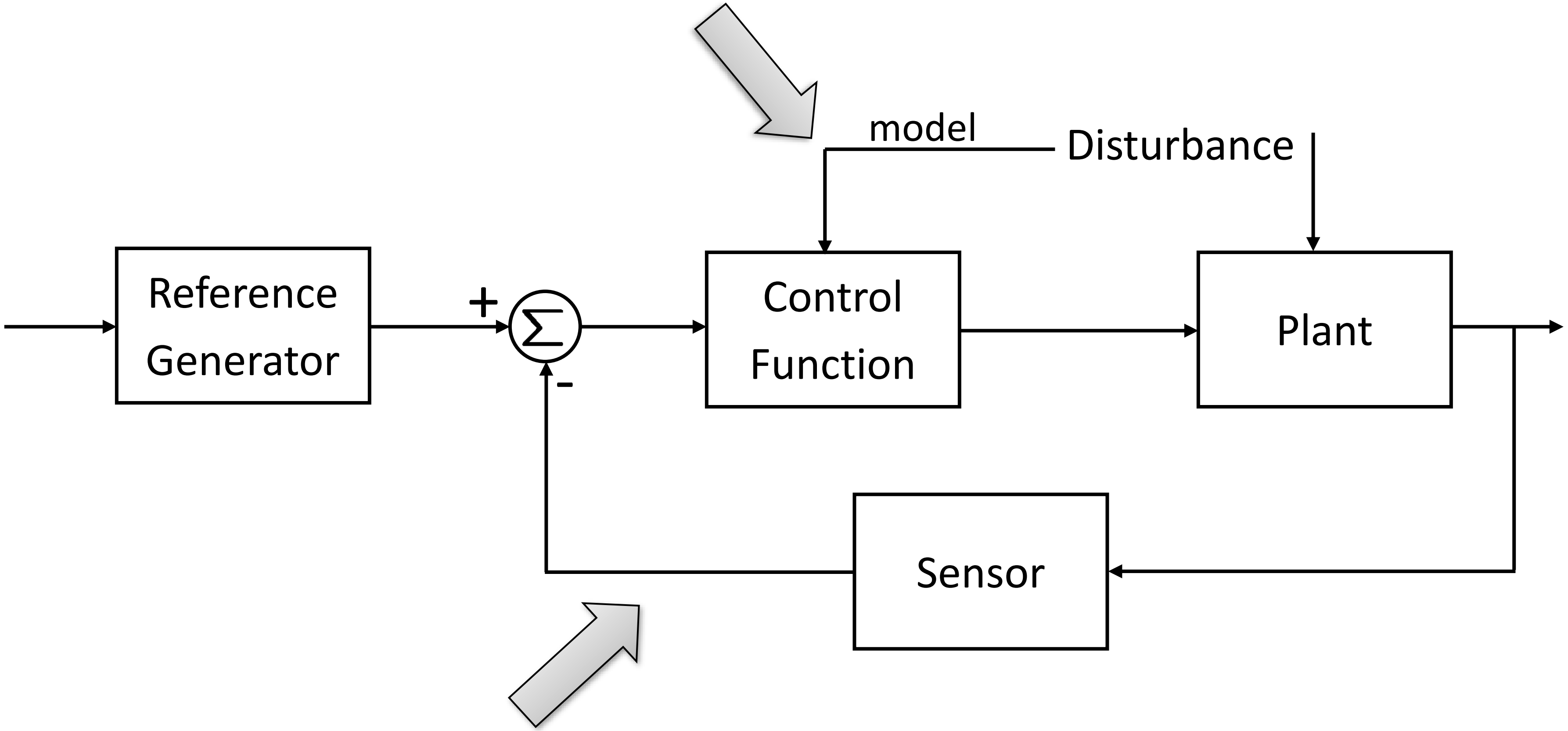


Let’s try...


CS225A Simulator
Control
Settings
float
Start
paste
Control Gains
kp:
400.0
kv:
40.0
Alternate parameters...
Output Files
Gather:
Start
Stop
data.mat
Plot:
Start
Stop
data.ps
Gripper
open
close
Status
q
0.0
copy
dq
0.0
copy
tau
-58.1
x
0.0
copy
F
0.00
0.00
0.00
0.00
0.00
0.00
1-DOF
FLOAT
EXIT
robottype 0
Simulator
18.70

A Simple Controller [Regler]

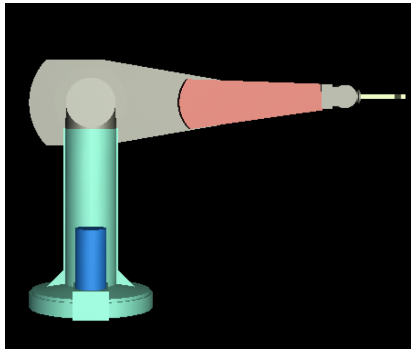
Feed-forward

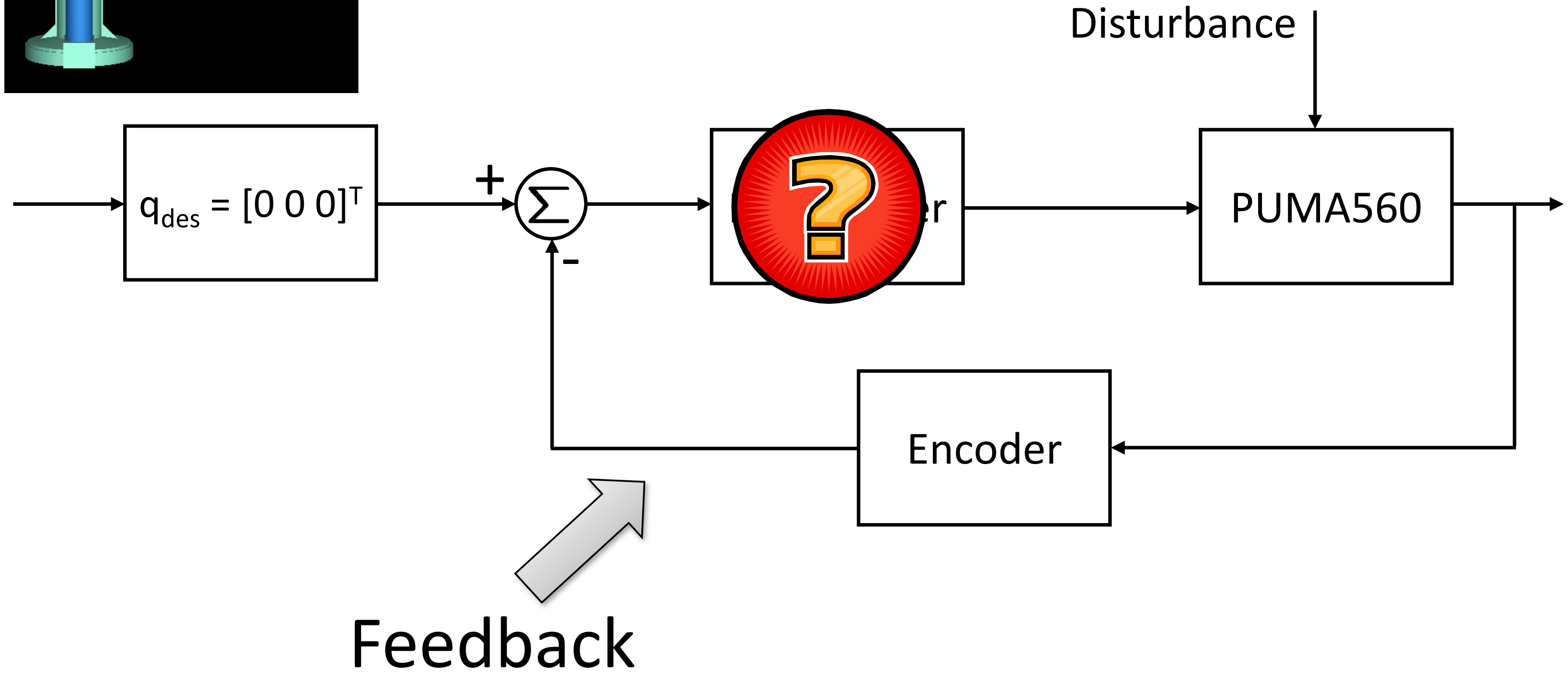


Feedback (we used proportional feedback or P control)

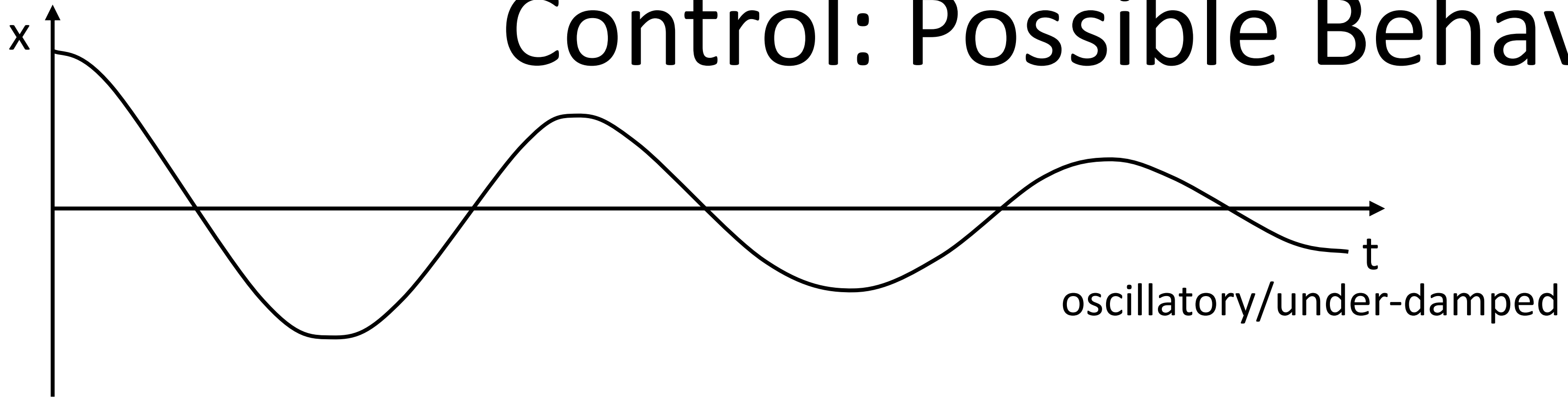
(also: closed loop – as opposed to open loop)

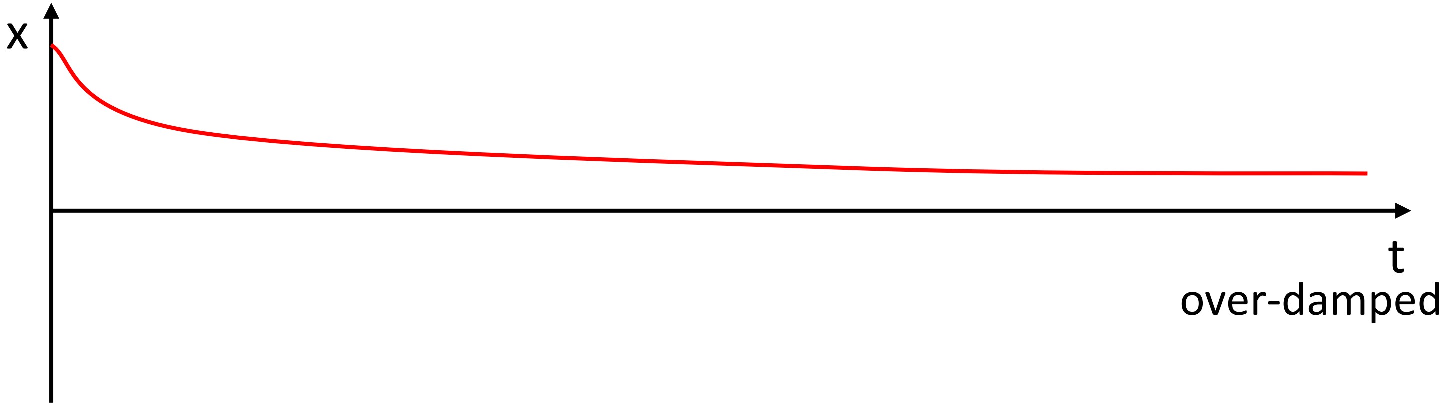
A Proportional Controller

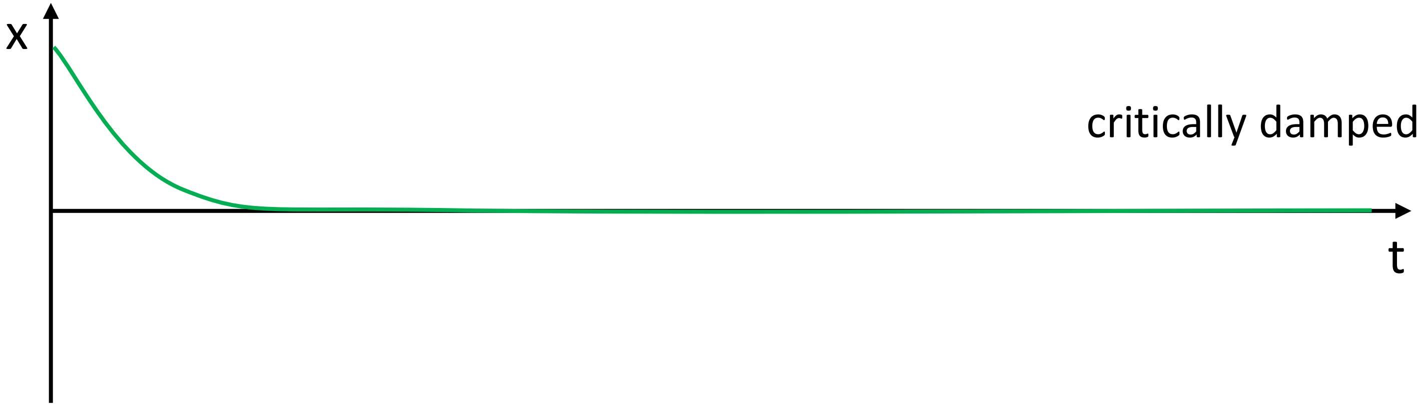




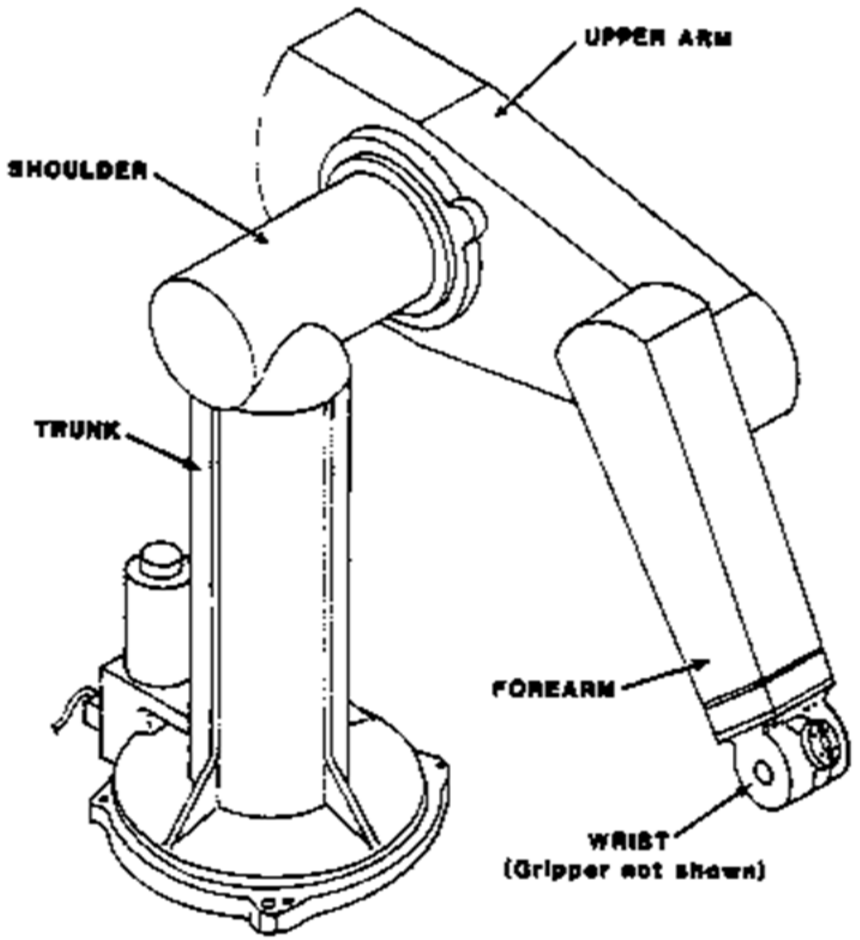
Control: Possible Behaviors







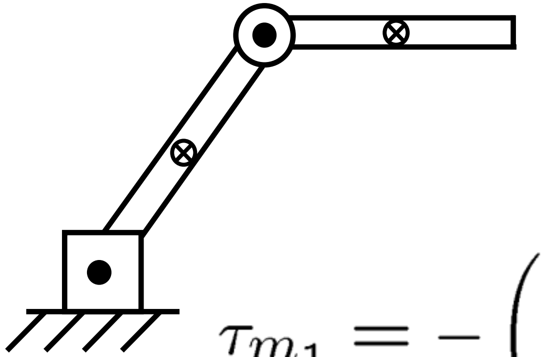
Possible Disturbances



- Gravity
- Inertia
- Centrifugal forces
- Coriolis effect
- Gears
- Actuator
- Friction
- Grasped object
- Contact
- …

What we need to solve the problem

- Fixed parameters of the robot
  - Kinematic
  - Dynamic
- Changing parameters
- Then we can:
  - Compensate gravity (feed-forward)
  - Reject error (feedback, P-controller)
- But: what happens when the robot moves?



$$\tau_{m_1} = -\begin{pmatrix} l\, c_1\, \mathbf{g}\, m_1 \\ 0 \end{pmatrix}$$

Disclaimer

These slides are intended as presentation aids for the lecture. They contain information that would otherwise be to difficult or time-consuming to reproduce on the board. But they are incomplete, not self-explanatory, and are not always used in the order they appear in this presentation. As a result, these slides should not be used as a script for this course. I recommend you take notes during class, maybe on the slides themselves. It has been shown that taking notes improves learning success.

Reading for this set of slides

- Craig – Intro to Robotics (3rd Edition)
  – Chapter 1

Please note that this set of slides is intended as support for the lecture, not as a stand-alone script. If you want to study for this course, please use these slides in conjunction with the indicated chapters in the text books. The textbooks are available online or in the TUB library (many copies that can be checked out for the entire semester. There are also some aspects of the lectures that will not be covered in the text books but can still be part of the homework or exam. For those It is important that you attend class or ask somebody about what was covered in class.

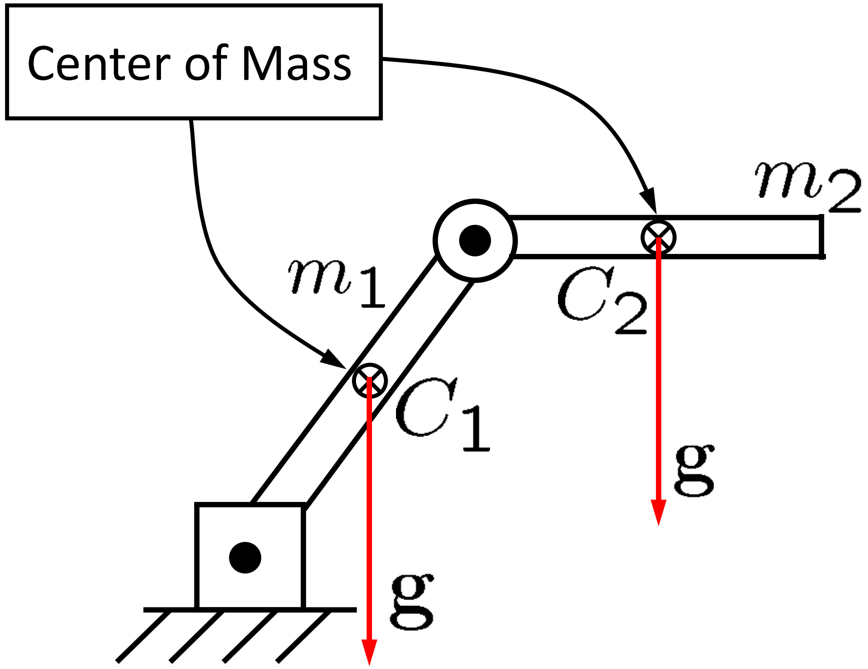
Robotics

A model of gravity based on simple physics

TU Berlin

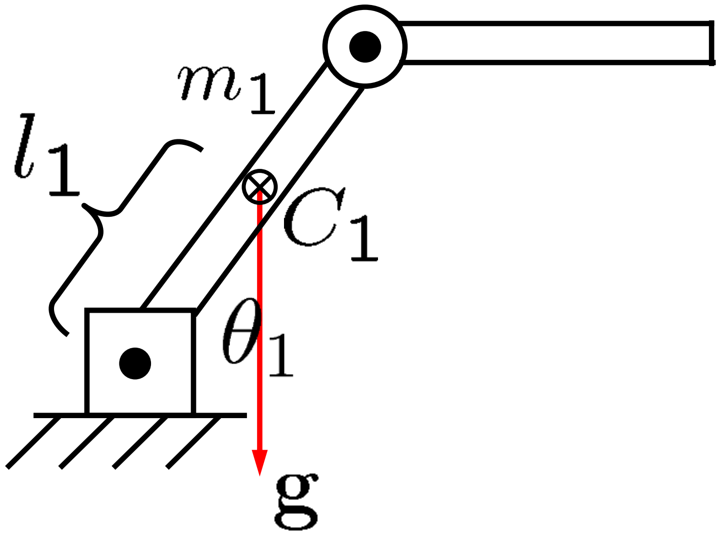
Oliver Brock

Gravity



$c_1 = \cos(\theta_1)$

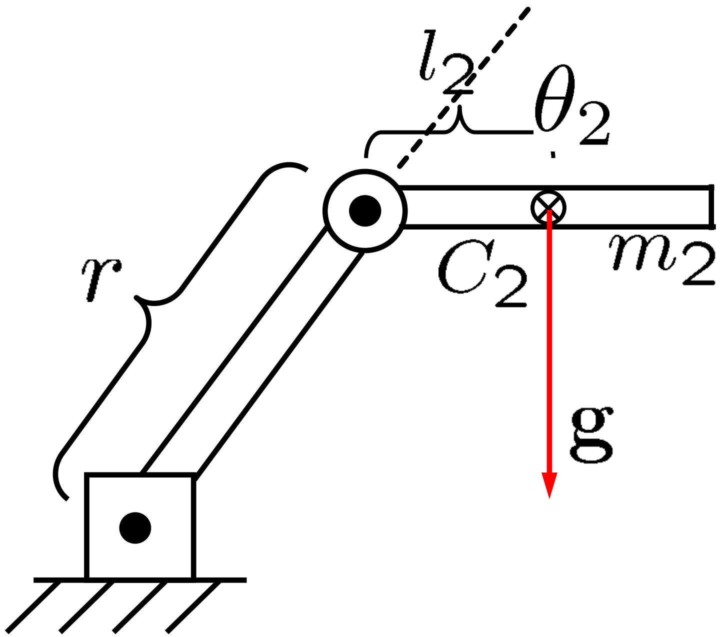
Gravity: Link 1



$$\tau_{m_1} = -\begin{pmatrix} l_1\ c_1\ \mathbf{g}\ m_1 \\ 0 \end{pmatrix}$$

$c_{12} = \cos(\theta_1 + \theta_2)$

Gravity: Link 2



$$\tau_{m_1} = -\begin{pmatrix} l_1\ c_1\ \mathbf{g}\ m_1 \\ 0 \end{pmatrix}$$

$$\tau_{m_2} = -\begin{pmatrix} (r\ c_1 + l_2\ c_{12})\ \mathbf{g}\ m_2 \\ l_2\ c_{12}\ \mathbf{g}\ m_2 \end{pmatrix}$$

$$\tau_g = \tau_{m_1} + \tau_{m_2} = -\begin{pmatrix} (r\ c_1 + l_2\ c_{12})\ m_2 + l_1\ c_1\ m_1 \\ l_2\ c_{12}\ m_2 \end{pmatrix} \mathbf{g}$$

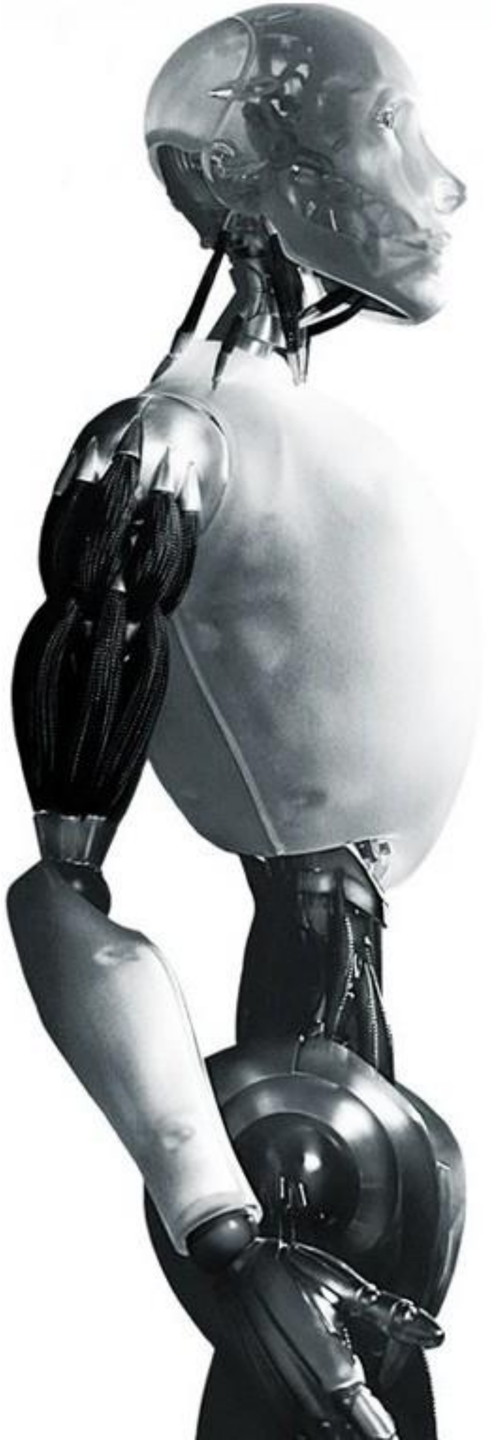
Disclaimer

These slides are intended as presentation aids for the lecture. They contain information that would otherwise be to difficult or time-consuming to reproduce on the board. But they are incomplete, not self-explanatory, and are not always used in the order they appear in this presentation. As a result, these slides should not be used as a script for this course. I recommend you take notes during class, maybe on the slides themselves. It has been shown that taking notes improves learning success.

Reading for this set of slides

- Craig – Intro to Robotics (3rd Edition)
  - Chapter 9.1 – 9.4

Please note that this set of slides is intended as support for the lecture, not as a stand-alone script. If you want to study for this course, please use these slides in conjunction with the indicated chapters in the text books. The textbooks are available online or in the TUB library (many copies that can be checked out for the entire semester. There are also some aspects of the lectures that will not be covered in the text books but can still be part of the homework or exam. For those It is important that you attend class or ask somebody about what was covered in class.



Robotics

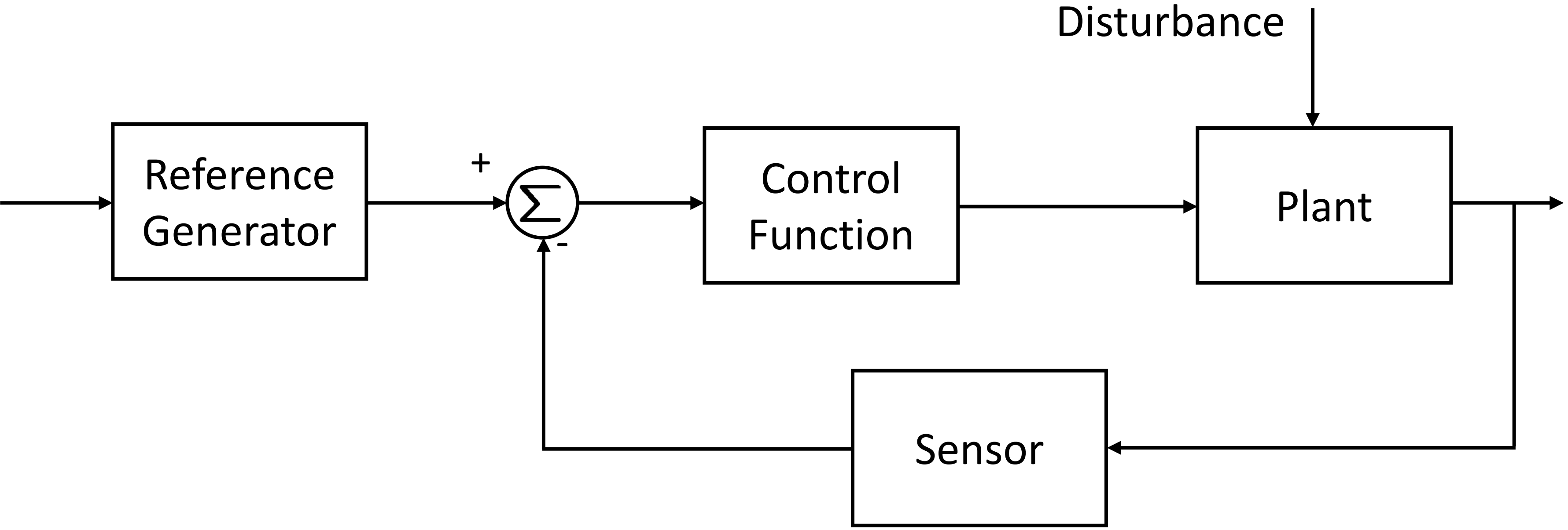
First Steps in Control

TU Berlin

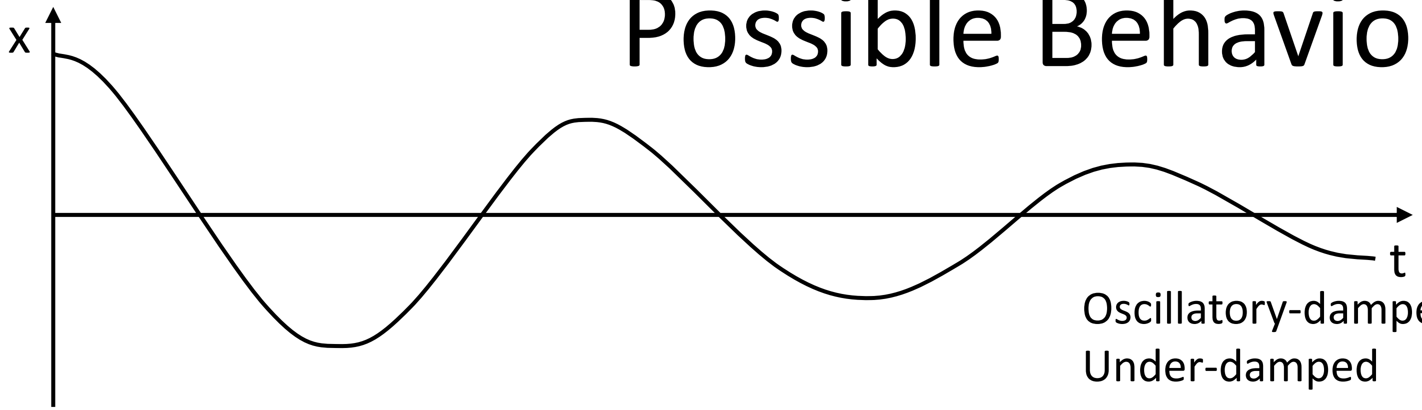
Oliver Brock

Control

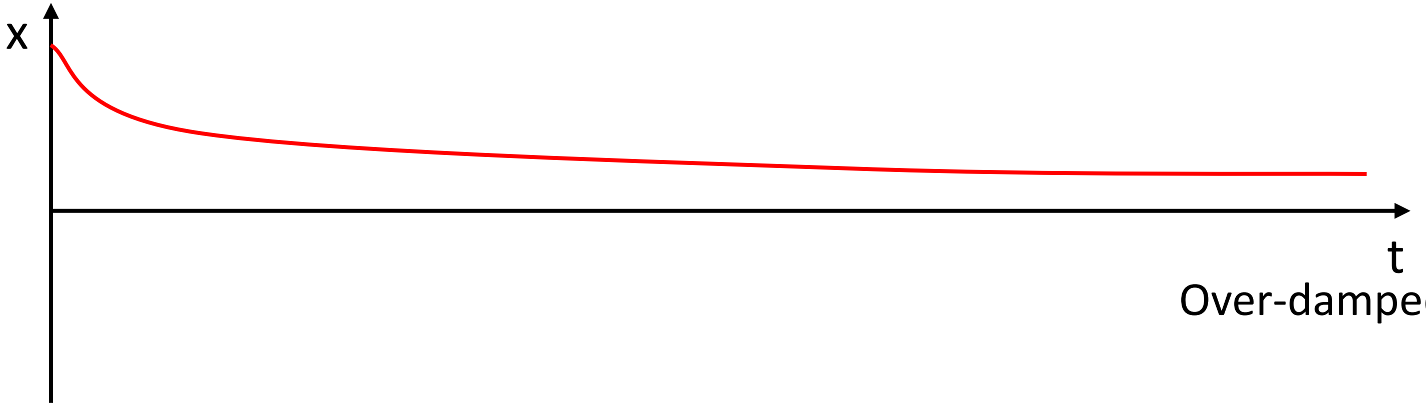
- **Control** is the process of causing a *system variable* to conform to some desired value, called a *reference value*.



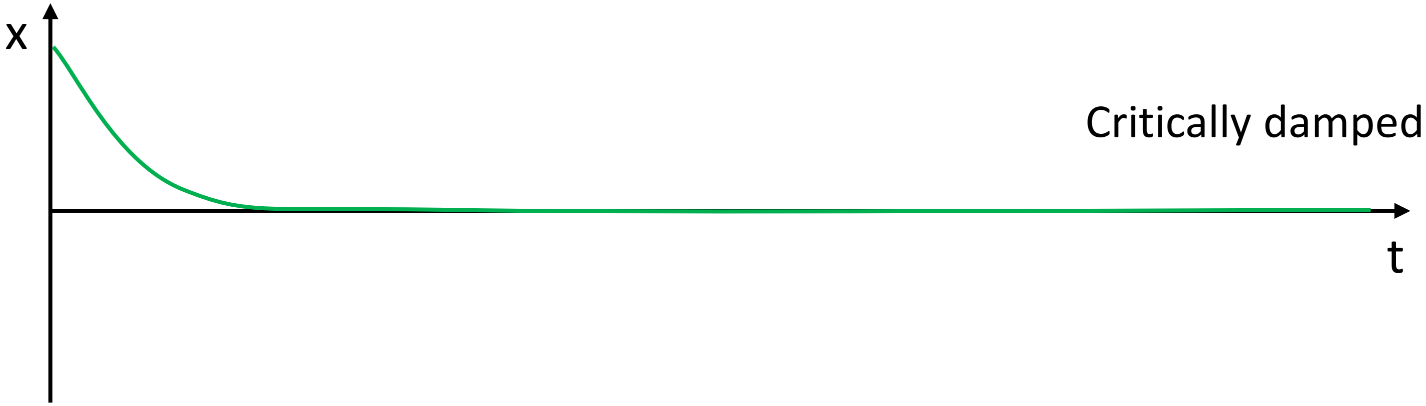
Possible Behaviors



Oscillatory-damped
Under-damped

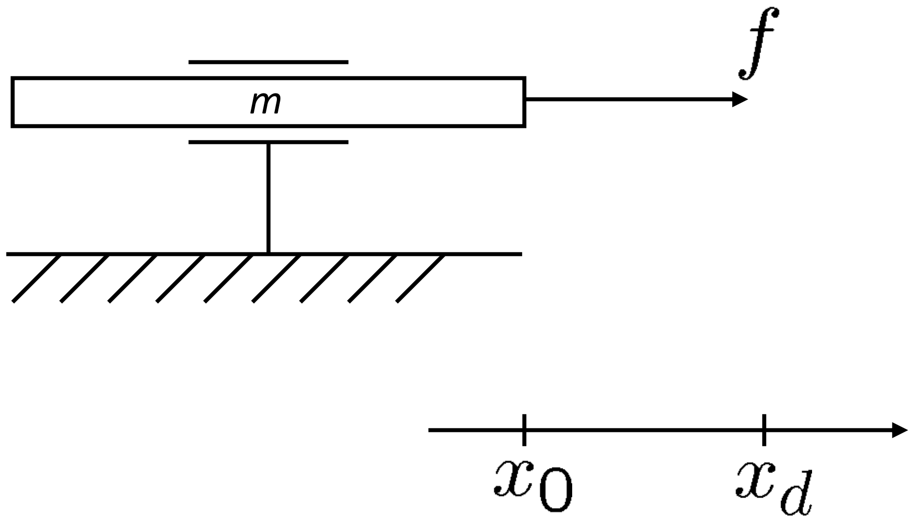


Over-damped



Critically damped

Simplest Case: Prismatic Joint

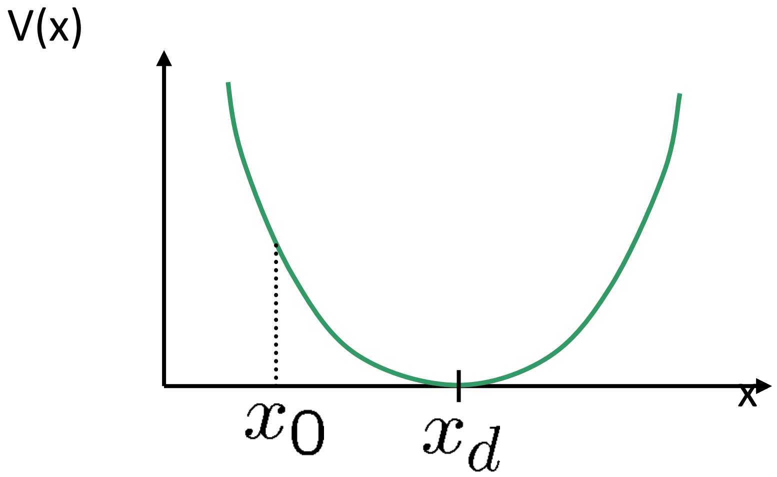


Proportional Control

Idea: apply force proportional to error

$$f = -\boxed{k_p}(x - x_d)$$

position gain

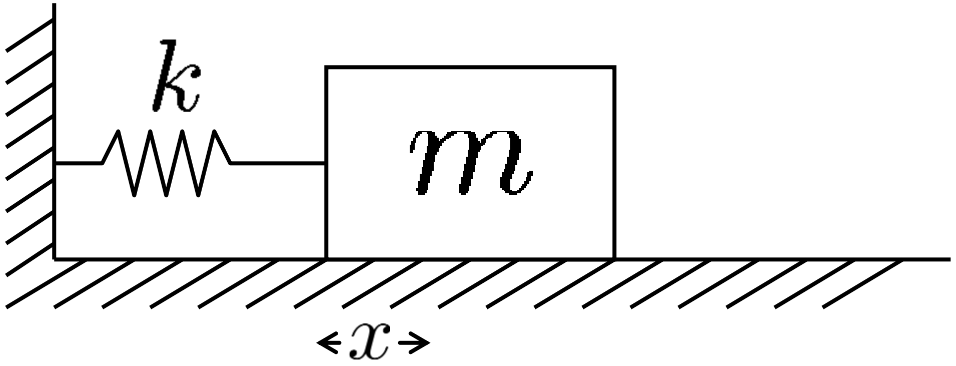


$$V(x) = \frac{1}{2} k_p (x - x_d)^2$$

$$\mathbf{F} = -\nabla V(x) = -\frac{\partial V}{\partial x}$$

Let’s start simple…

Conservative System / Simple Harmonic Oscillator



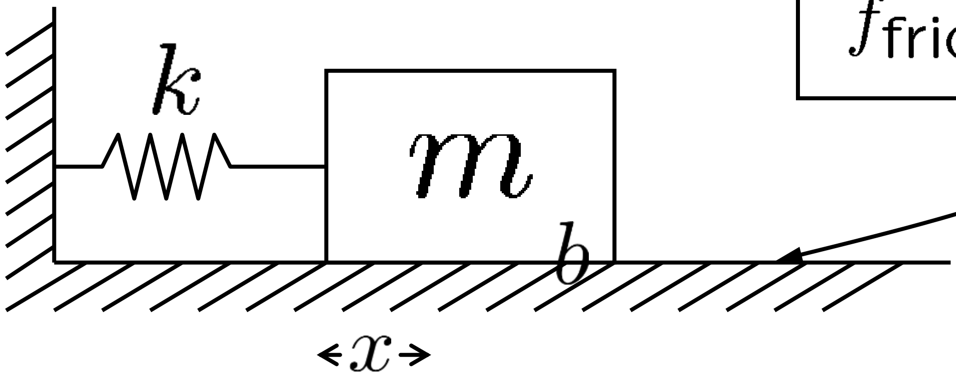
Equation of motion

$$m\ddot{x} + kx = 0$$

Dissipative System: Add Friction

viscous friction

$$f_{\text{friction}} = -b\dot{x}$$



Equation of motion

$$m\ddot{x} + b\dot{x} + kx = 0$$

Introduction of Dissipation

Idea: apply force opposing velocity

this leads to dissipation (or dampening)

$$f = -k_p x - \underbrace{\boxed{k_v}}_{\text{velocity gain}} \dot{x}$$

PD Control: Adding an Actuator

$$m\ddot{x} + b\dot{x} + kx = 0$$

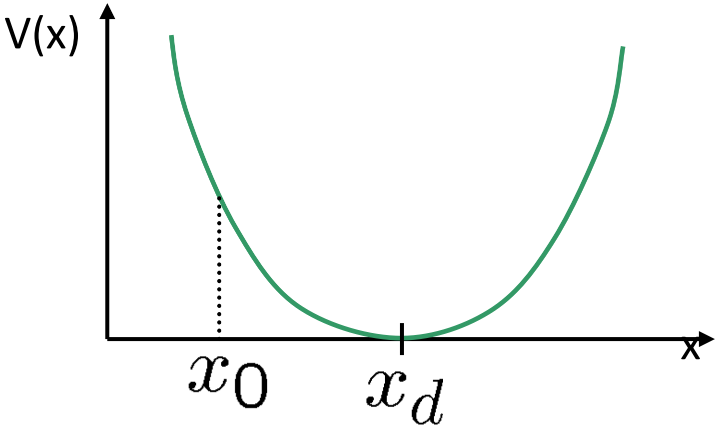
$$f = m\ddot{x} + b\dot{x} + kx$$

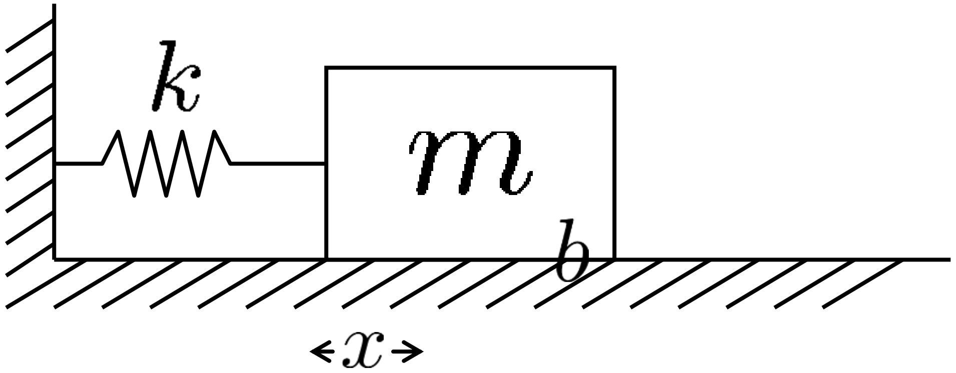
$$f = -k_p\,x - k_v\,\dot{x}$$

$$m\ddot{x} + b\dot{x} + kx = -k_p x - k_v \dot{x}$$

$$m\ddot{x} + (b + k_v)\dot{x} + (k + k_p)x = 0$$

Summary





$$\mathbf{F} = -\nabla V(x) = -\frac{\partial V}{\partial x}$$

$$\tau = -k_p(p - q_d) - k_v \dot{q}$$

Robotics – Tutorial for Assignment #1

By **Előd Páll**

The Puma560 Simulator

- Software that simulates the Kinematics, Dynamics, Friction etc. of the Puma560 robot
- *pumasim* binary or Virtual-Box
- Provides a GUI for controlling, monitoring and configuring the simulation

Running the simulator natively

- Known to work on Ubuntu 20.04 and 18.04
- Download the pumasimulator-xxxx.tgz

  tar –xzvf pumasimulator-xxxx.tgz
  cd pumasimulator
- Read the *Readme.md* for installation instructions

Running the simulator with a VM image

- Install Oracle **Virtualbox 6.1**
  http://virtualbox.org/
- Download the Virtual Machine image (.ova) from ISIS
  - Tip: Do it on the campus net (ca. 2.6GB)
- Start the machine
- Login: student
- Password: student
- Open a terminal and type *pumasim*

Simulator internals

- Controller are called every 2 ms
- Predefined names
- The pumasim executable does not contain controllers, but loads them from the shared library *controlDLL.so*
- Pumasim first looks for the library in the current working directory, then in /opt/pumasim
- You only need to compile controlDLL.so

control.cpp

- Here you implement your robot controller
- **Important:**
  - File must also be compatible on the Real-Time-PC running QNX.
- `init…()` functions are called when you click on „Start" (controller).
- `…control()` functions are called periodically in the servo loop with 500Hz.
- A lot of global variables are declared via the structure *gv*: they contain the simulator/robot's state

data.mat

- Plain ASCII text file
- Each line is a timepoint:

time q(1..n) dq(1..n) qd(1..n) tau(1..n) x(1..m) dx(1..m) xd(1..m)

- n = DOF
- m = 7 in „6-DOF (quaternions)“ mode
  else m = DOF
- You can import it into Excel, Matlab, gnuplot, Octave, matplotlib, a.s.o. and make nice graphs

gains_*.txt

- Text file containing separate gains for all controllers for a specific robot mode
  - gains_1.txt = gains during 3DOF mode
  - gains_6.txt = gains during 6DOF quaternion mode
  - a.s.o.

- Please do not edit the text file manually, but use *Store gains* and *Load gains in the GUI*

Before you start coding

- **Read** *Notes and Restrictions on Coding.pdf* (available on ISIS)
- Information about available math library (vector, matrix, etc.), global variables, etc.

P-controller

- Important variables for the P-controller in the gv struct:

```
tau     : joint torques
q       : joint position
kp      : position gains for the current control mode
qd      : desired joint position
```

- You can tune your controller via the GUI
- You can visualize signals by writing them to a text file (.mat)

Compile System for controlDLL

- Cmake based compile template:

```
cd 1/
mkdir build
cd build
cmake ..
make
```

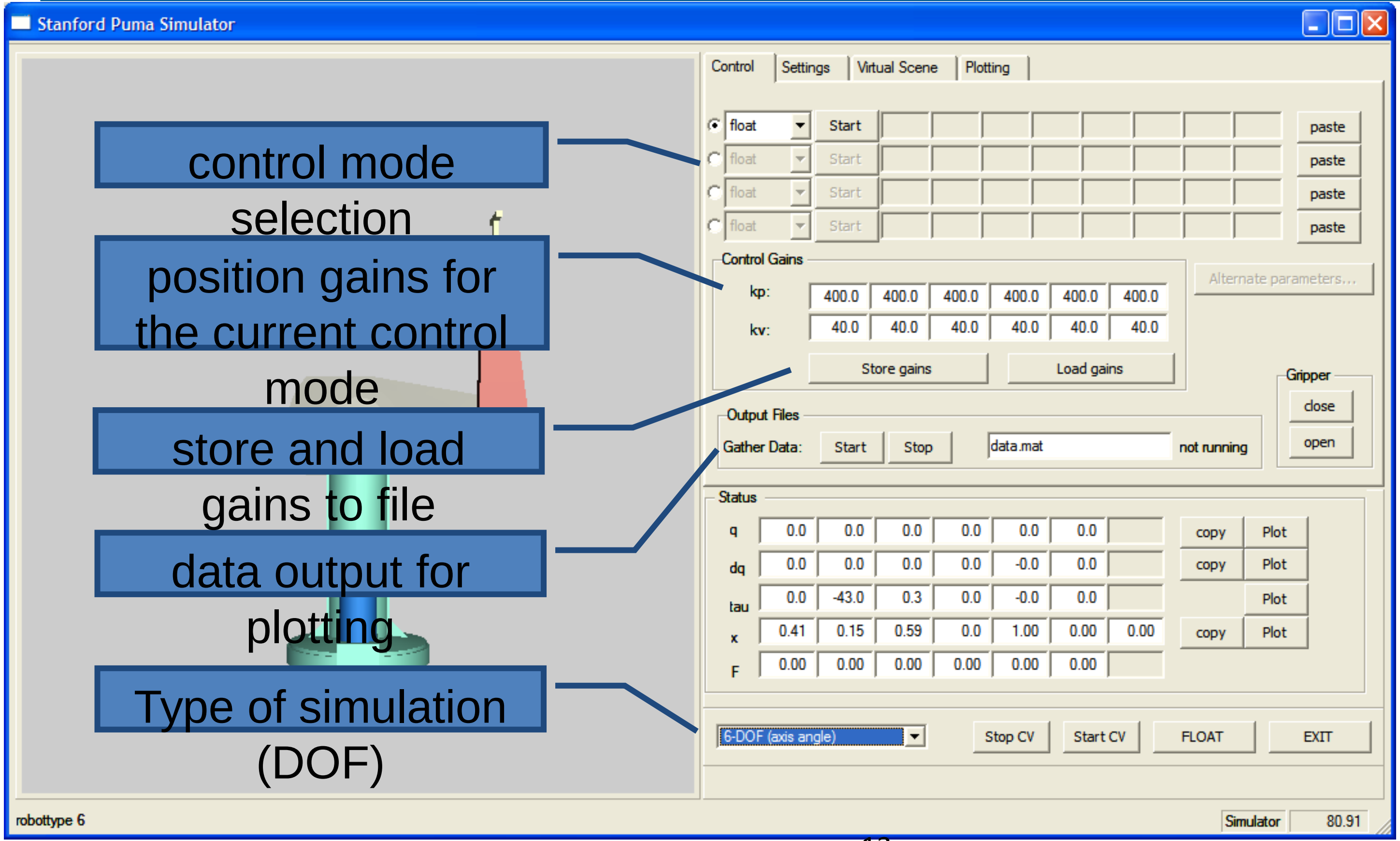
- This creates a controlDLL.so from control.cpp
- To use your controller, call *pumasim* in the *build/* directory:

```
pumasim
```

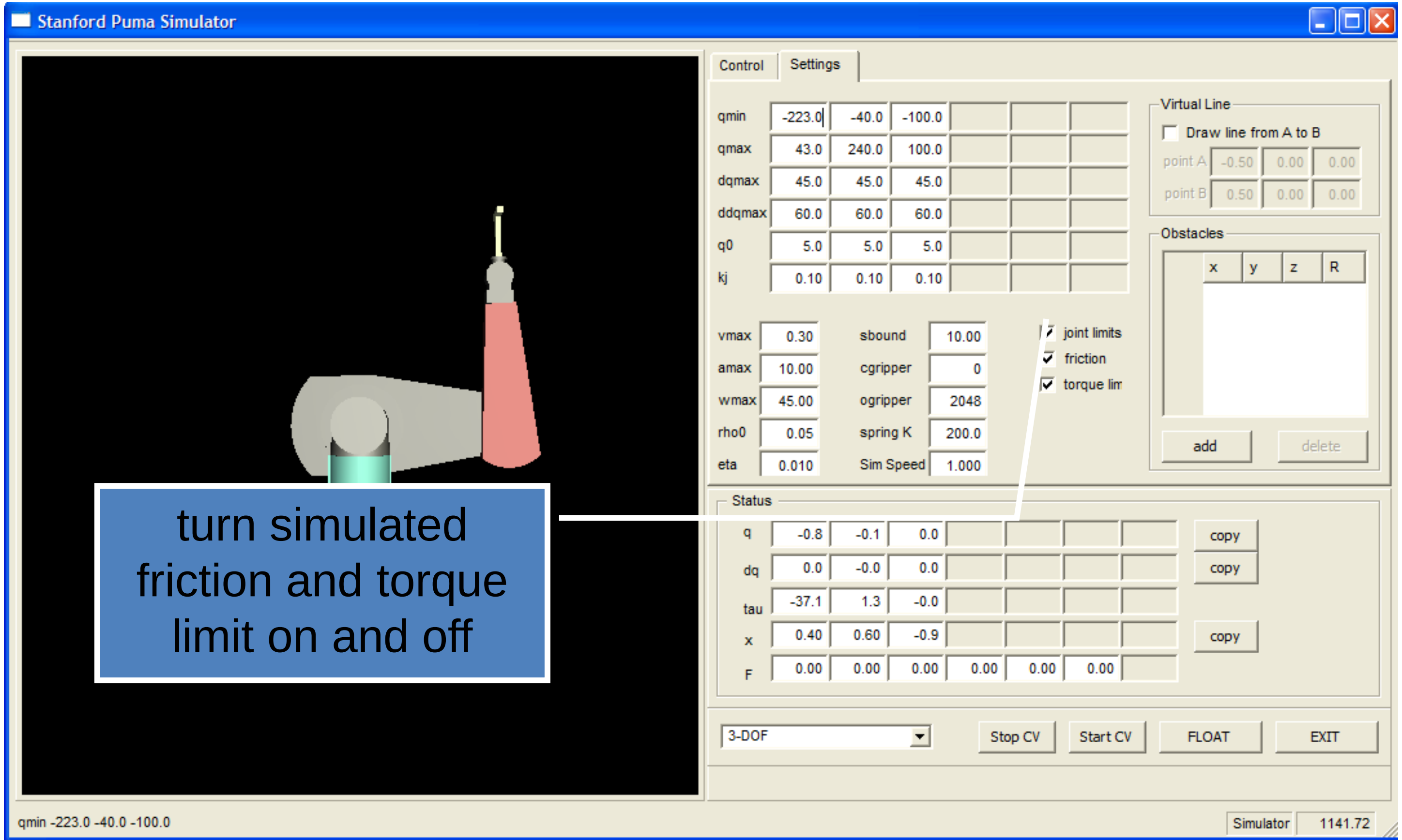
Q&A

- ISIS discussion forum
- teaching@robotics.tu-berlin.de

Puma Simulator



Puma Simulator Settings



turn simulated friction and torque limit on and off