THE UNIVERSITY of EDINBURGH
informatics

Advanced Robotics

Dynamics I

Subramanian Ramamoorthy
School of Informatics
University of Edinburgh

Dynamics vs. Kinematics

- So far we assumed that we can generate any $\mathbf{v_q}$ on our robot (eg when looking at forward kinematics)

- However this is rarely the case, eg:
  - A flying airplane: You cannot command it to hold still in the air or move straight up
  - A car: You cannot command it to move sideways
  - Your arm: You can’t command it to throw a ball with arbitrary velocity (force limits)
  - A torque-controlled robot: You cannot command it to instantaneously change velocity (infinite acceleration/torque)

Actuation of a robot

An actuator needs a model:

- **x** is the state of the actuator / robot
- **u** is the control input
- The state at any time depends on both the previous state and the control input:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t)$$

, with *f* assumed to be smooth

- Eg: when we looked at forward / inverse kinematics

$$\mathbf{x} = \mathbf{q}, \mathbf{u} = \mathbf{v_q}$$

$$f(\mathbf{q}, \mathbf{v_q}) = \mathbf{q} \oplus \mathbf{v_q}$$

Three classic models for a robot actuator

- Velocity source $\mathbf{x} = \mathbf{q}, \mathbf{u} = \mathbf{v_q}$

  Good approximation for hydraulic motors; good for electric actuators only in certain condition (eg industrial manipulators, not legged robots)

Three classic models for a robot actuator

- Velocity source $\mathbf{x} = \mathbf{q}, \mathbf{u} = \mathbf{v_q}$

  Good approximation for hydraulic motors; good for electric actuators only in certain condition (eg industrial manipulators, not legged robots)

- Acceleration / force source $\mathbf{x} = (\mathbf{q}, \mathbf{v_q}), \mathbf{u} = \dot{\mathbf{v_q}}$

  Good approximation for electric motors if large contact forces are not involved.

Three classic models for a robot actuator

- Velocity source $\mathbf{x} = \mathbf{q}, \mathbf{u} = \mathbf{v_q}$
  Good approximation for hydraulic motors; good for electric actuators only in certain condition (eg industrial manipulators, not legged robots)

- Acceleration / force source $\mathbf{x} = (\mathbf{q}, \mathbf{v_q}), \mathbf{u} = \dot{\mathbf{v_q}}$
  Good approximation for electric motors if large contact forces are not involved.

- Torque source $\mathbf{x} = (\mathbf{q}, \mathbf{v_q}), \mathbf{u} = \tau$
  Good approximation for electric motors. Assumption is that torque is proportional to current. However, gear reductions introduce unmodeled terms that we need to account for.

Outline

- We discuss the following three topics today:
  - 1D point mass
  - A ‘general’ dynamic robot (□ Dynamics II)
  - Joint space control

- For now we assume that the robot is fully actuated and that $\mathbf{v_q} = \dot{\mathbf{q}}$

  (ie velocity and configuration space have the same dimension)

- We also assume motors are equipped with accurate **position** sensors (i.e. we know **q** accurately)

Exercise on the Board

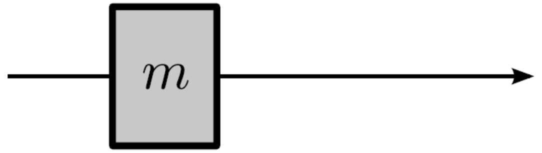
Discuss problem formulation and modelling assumptions associated with moving an object, along a computed trajectory, and controlling against deviations

- How to phrase the questions?

- How does analysis support design?

Simplest possible case: 1D point mass

- no gravity, no friction



- **State** $x(t) = (q(t), \dot{q}(t))$ is described by:
  - position $q(t) \in \mathbb{R}$
  - velocity $\dot{q}(t) \in \mathbb{R}$
- The **controls** $u(t)$ is the force we apply on the mass point
- The **system dynamics** is:

$$\ddot{q}(t) = u(t)/m$$

1D Mass

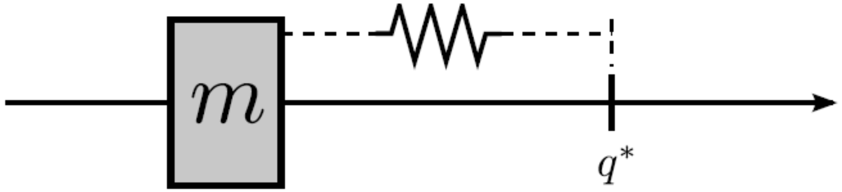
- Given current q_t, what control u_t to get closer to desired position q^*?

1D Mass: Proportional Control

- Given current q_t, what control u_t to get closer to desired position q^*?

- Consider an applied force that is an input proportional to the “error”:
  (difference here is that **u** is a force and not a velocity)

$$u = K_p \left(q^* - q\right)$$



- You can picture a spring attached to q* that pulls the mass towards it. What happens in the absence of friction ?

1D Mass: Closed-loop Dynamics

$$m\,\ddot{q} = u = K_p\,(q^* - q)$$

q=q(t) is a function of time, this is a second order differential eq.

- Solution: assume $q(t) = a + be^{\omega t}$
  (an "non-imaginary" alternative would be $q(t) = a + b\,\epsilon^{-\lambda t}\ \cos(\omega t)$)

$$m\,b\,\omega^2\,e^{\omega t} = K_p\,q^* - K_p\,a - K_p\,b\,e^{\omega t}$$

$$(m\,b\,\omega^2 + K_p\,b)\,e^{\omega t} = K_p\,(q^* - a)$$

$$\Rightarrow (m\,b\,\omega^2 + K_p\,b) = 0\ \wedge\ (q^* - a) = 0$$

$$\Rightarrow\ \omega = i\sqrt{K_p/m}$$

$$q(t) = q^* + b\,e^{i\sqrt{K_p/m}\ t}$$

This is an oscillation around q^* with amplitude $b = q(0) - q^*$ and frequency $\sqrt{K_p/m}$!

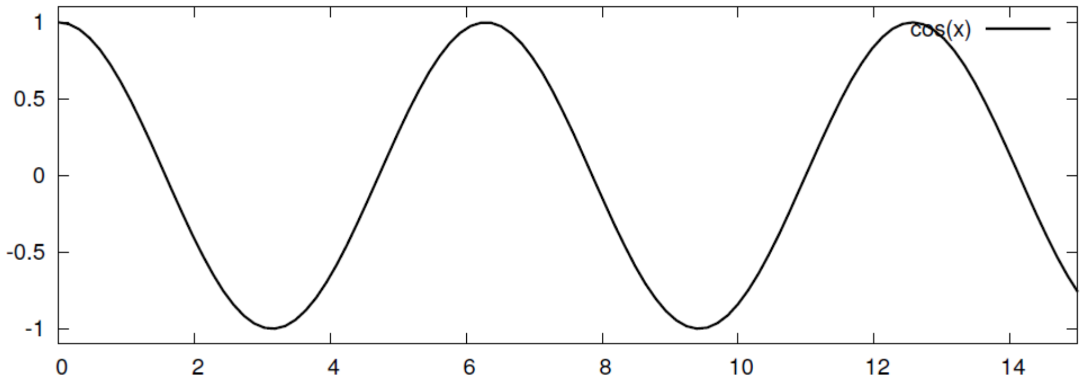
1D Mass: Closed-loop Dynamics

What's the effect?

$$m\,\ddot{q} = u = K_p\,(q^* - q)$$

$$q(t) = q^* + b\,e^{i\sqrt{K_p/m}\;t}$$

This is an oscillation around q^* with amplitude $b = q(0) - q^*$ and frequency $\sqrt{K_p/m}$!

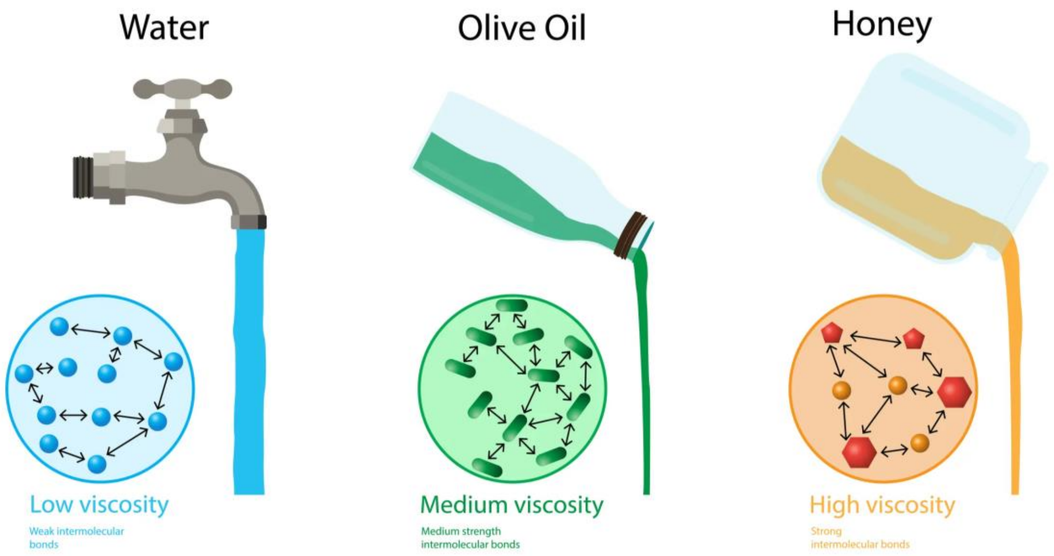


1D Mass: Damping forces

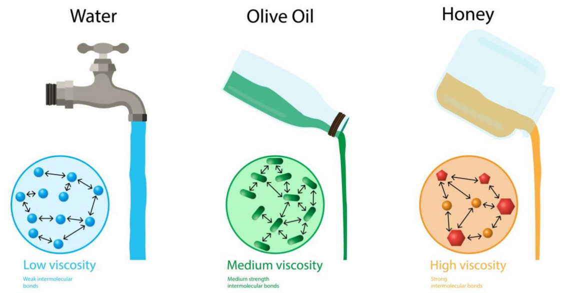
Can we shape the dynamics further?

“Pull less, when we’re heading the right direction already:”

“Damp the system:”



1D Mass: Damping forces

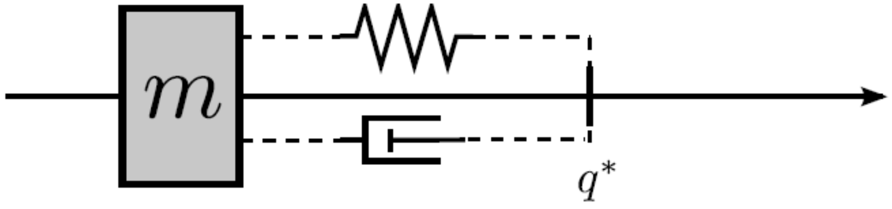


Can we shape the dynamics further?

"Pull less, when we're heading the right direction already:"
"Damp the system:"

$$u = K_p(q^* - q) + K_d(\dot{q}^* - \dot{q})$$

$\dot{q}^*$ is a desired goal velocity
For simplicity we set $\dot{q}^* = 0$ in the following.



1D Mass: Damping forces

What is the effect? $m\ddot{q} = u = K_p(q^* - q) + K_d(0 - \dot{q})$

- Solution: again assume $q(t) = a + be^{\omega t}$

$$m\, b\, \omega^2\, e^{\omega t} = K_p\, q^* - K_p\, a - K_p\, b\, e^{\omega t} - K_d\, b\, \omega e^{\omega t}$$

$$(m\, b\, \omega^2 + K_d\, b\, \omega + K_p\, b)\, e^{\omega t} = K_p\, (q^* - a)$$

$$\Rightarrow (m\, \omega^2 + K_d\, \omega + K_p) = 0 \;\wedge\; (q^* - a) = 0$$

$$\Rightarrow \omega = \frac{-K_d \pm \sqrt{K_d^2 - 4mK_p}}{2m}$$

$$q(t) = q^* + b\, e^{\omega\, t}$$

The term $-\frac{K_d}{2m}$ in ω is real $\leftrightarrow$ exponential decay (damping)

1D Mass: Damping forces

What's the effect?

$$q(t) = q^* + b\, e^{\omega\, t}\,, \qquad \omega = \frac{-K_d \pm \sqrt{K_d^2 - 4mK_p}}{2m}$$

- Effect of the second term $\sqrt{K_d^2 - 4mK_p}/2m$ in ω:

$K_d^2 < 4mK_p \quad \Rightarrow$ ω has imaginary part
oscillating with frequency $\sqrt{K_p/m - K_d^2/4m^2}$
$q(t) = q^* + be^{-K_d/2m\ t}\ e^{i\sqrt{K_p/m - K_d^2/4m^2}\ t}$

$K_d^2 > 4mK_p \quad \Rightarrow$ ω real
strongly damped

$K_d^2 = 4mK_p \quad \Rightarrow$ second term zero
only exponential decay

1D Mass: Concept of *damping ratio*

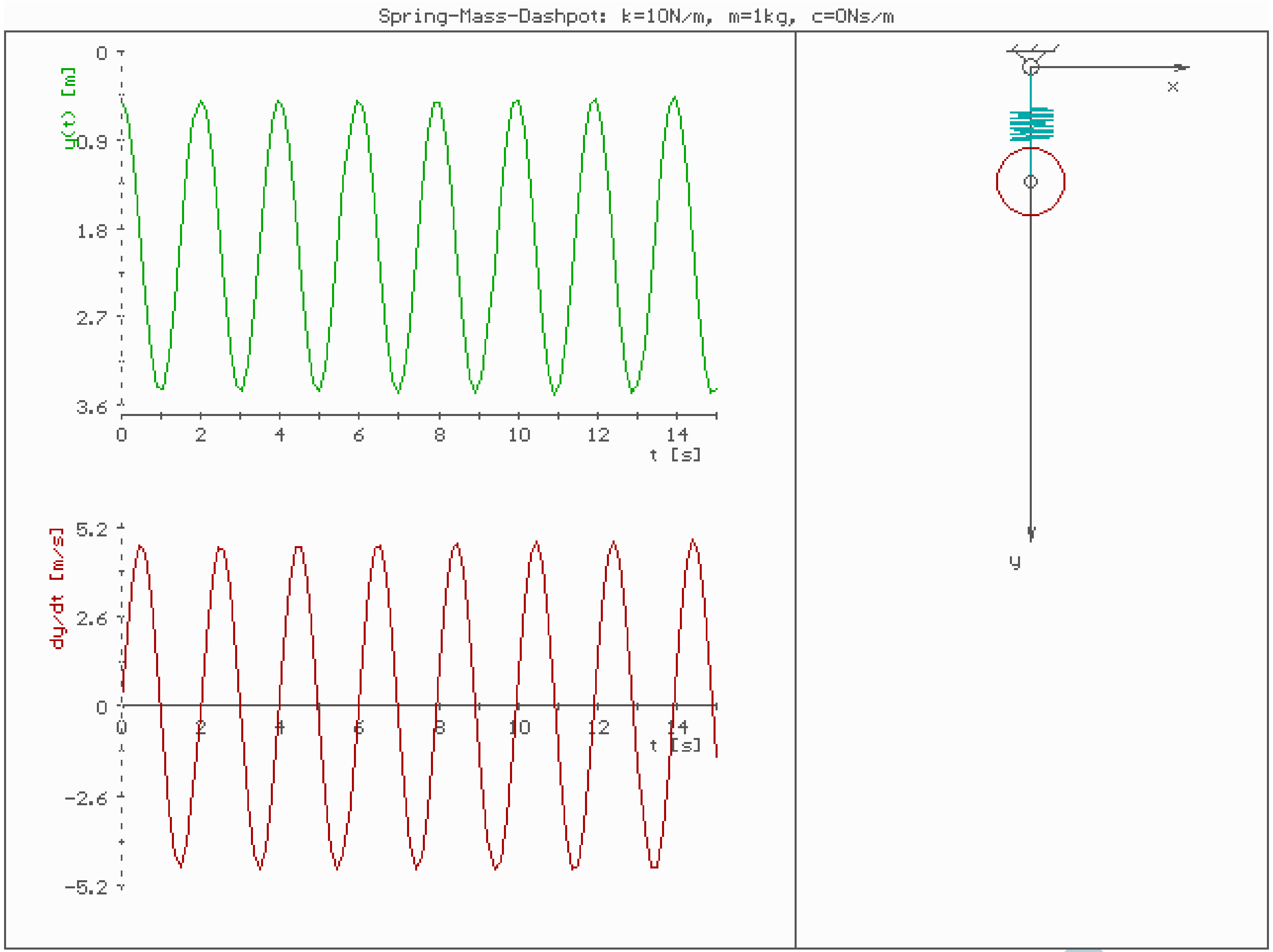
Alternative Parameterisation

Instead of specifying the *coefficients* *Kp* and *Kd*

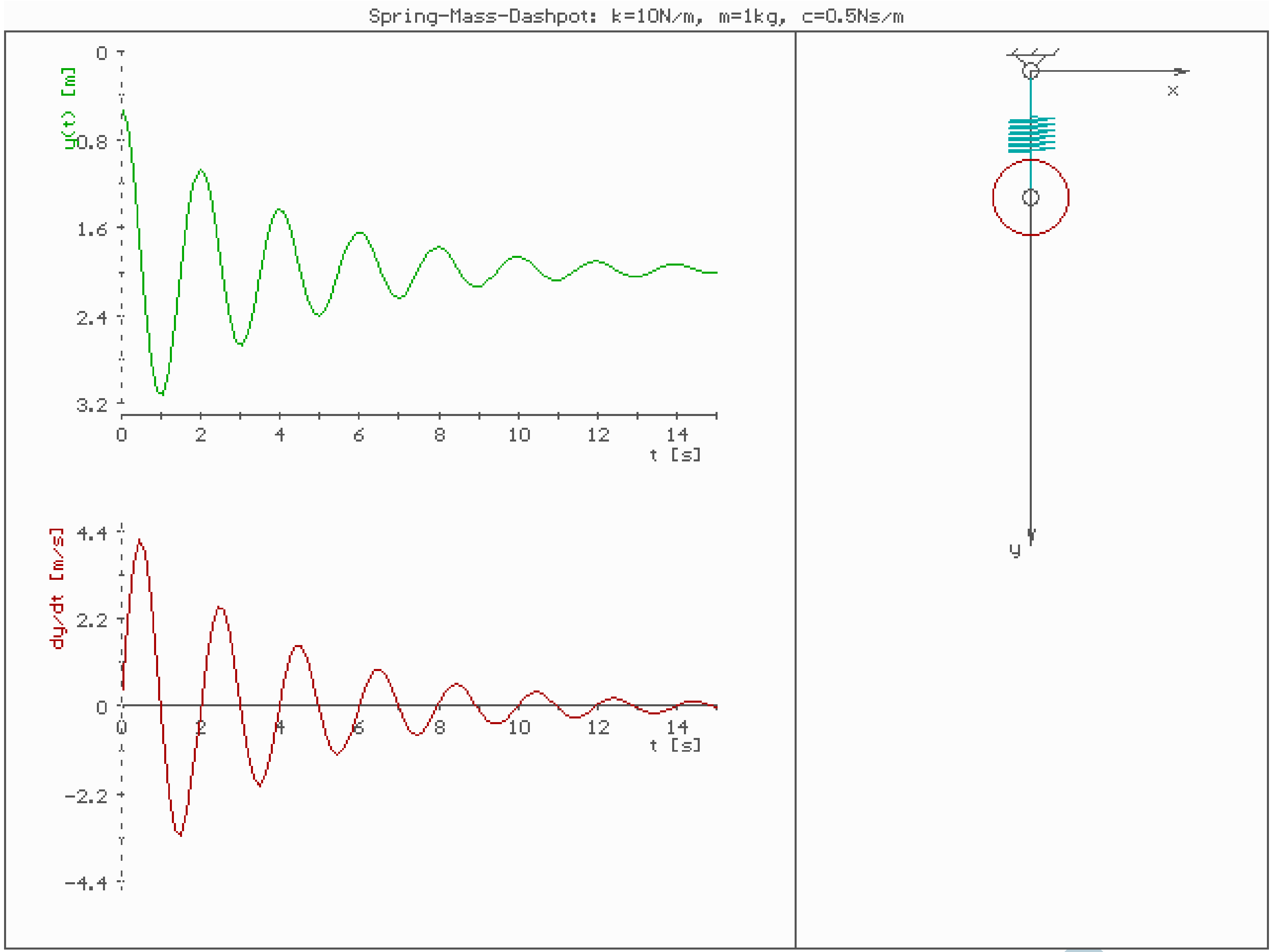
- "wave length" $\lambda = \frac{1}{\omega_0} = \frac{1}{\sqrt{K_p/m}}$, $K_p = m/\lambda^2$
- damping ratio $\xi = \frac{K_d}{\sqrt{4mK_p}} = \frac{\lambda K_d}{2m}$, $K_d = 2m\xi/\lambda$

$\xi > 1$: over-damped
$\xi = 1$: critically dampled
$\xi < 1$: oscillatory-damped

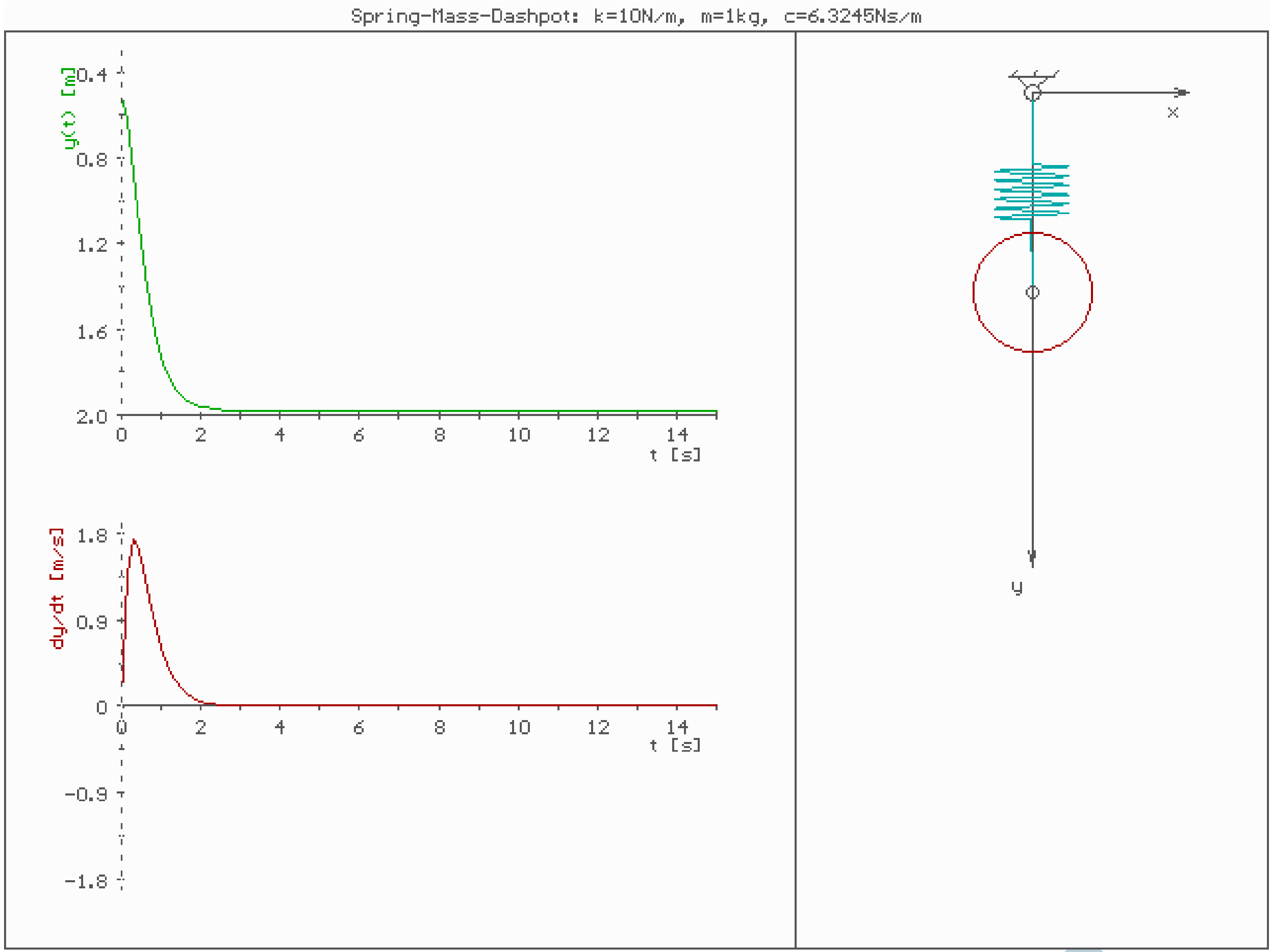
$$q(t) = q^* + be^{-\xi/\lambda\ t}\ e^{i\omega_0\sqrt{1-\xi^2}\ t}$$



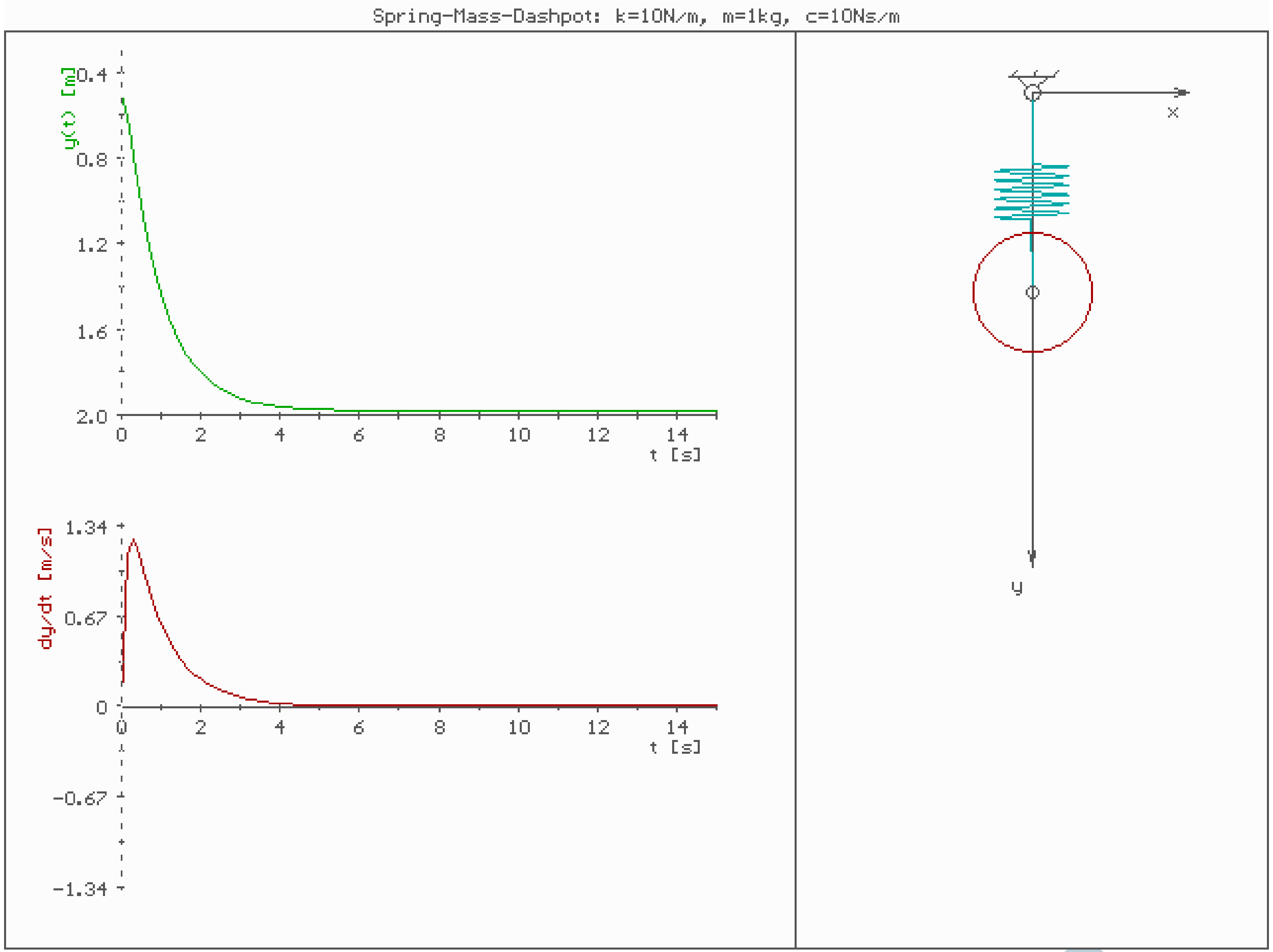
Spring-Mass-Dashpot: k=10N/m, m=1kg, c=0Ns/m
y(t) [m]
0
0.9
1.8
2.7
3.6
0
2
4
6
8
10
12
14
t [s]
dy/dt [m/s]
5.2
2.6
0
-2.6
-5.2
t [s]
x
y



Spring-Mass-Dashpot: k=10N/m, m=1kg, c=0.5Ns/m
y(t) [m]
0
0.8
1.6
2.4
3.2
0
2
4
6
8
10
12
14
t [s]
dy/dt [m/s]
4.4
2.2
0
-2.2
-4.4
0
2
4
6
8
10
12
14
t [s]
x
y



Spring-Mass-Dashpot: k=10N/m, m=1kg, c=6.3245Ns/m
y(t) [m]
0.4
0.8
1.2
1.6
2.0
0
2
4
6
8
10
12
14
t [s]
dy/dt [m/s]
1.8
0.9
0
-0.9
-1.8
0
2
4
6
8
10
12
14
t [s]
x
y



Spring-Mass-Dashpot: k=10N/m, m=1kg, c=10Ns/m
y(t) [m]
0.4
0.8
1.2
1.6
2.0
0
2
4
6
8
10
12
14
t [s]
dy/dt [m/s]
1.34
0.67
0
-0.67
-1.34
x
y

Adding another term to this Spring-damper system

$$u = K_p(q^* - q) + K_d(\dot{q}^* - \dot{q}) + K_i \int_{s=0}^{t} (q^*(s) - q(s))\, ds$$

- **PID control**
  - – Proportional Control ("Position Control")
    $f \propto K_p(q^* - q)$
  - – Derivative Control ("Damping")
    $f \propto K_d(\dot{q}^* - \dot{q}) \quad (\dot{x}^* = 0 \rightarrow$ damping)
  - – Integral Control ("Steady State Error")
    $f \propto K_i \int_{s=0}^{t} (q^*(s) - q(s))\, ds$

1D Mass: Summary

- ❑ Dynamics of a 1D mass-spring-damper system, a spring and a damper added to a point mass, i.e. spring and damping forces (aka PD controller)
- ❑ Resultant force acting on the system in a linear ‘control law’

$$\pi : \ (q, \dot{q}) \mapsto u = K_p(q^* - q) + K_d(\dot{q}^* - \dot{q})$$

(linear in the *dynamic system state* $x = (q, \dot{q})$)

- ❑ With such simple linear rules, we can modulate the dynamic response of the system by tuning the ‘strength’ of spring and damper (ie, PD gains in the PD ‘control law’)

- ❑ *trade-off: there is no optimality criterion supporting such rules and the resulting motions (hence, we may be able to do better)