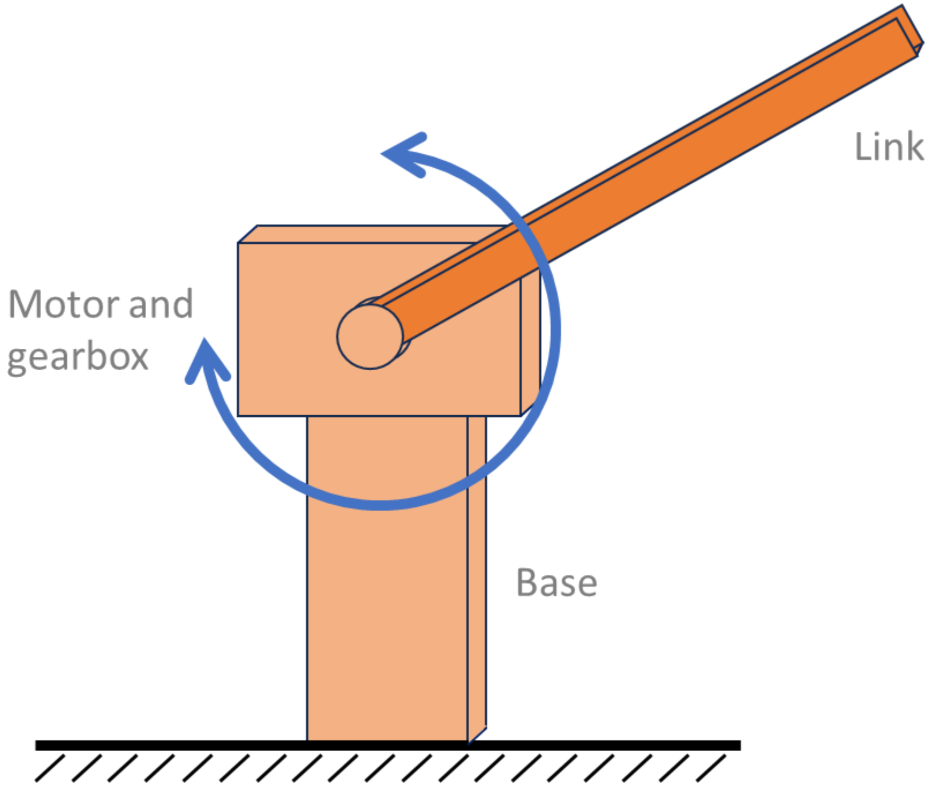
Manipulator Dynamics & Control:

Single Link Manipulator

{Learn, Create, Innovate};

Introduction

- In this lecture a model of the dynamics and control of a single link manipulator (SLM) will be developed.
    1. Introduction
    2. Dynamical behaviour of a SLM
    3. Linear analysis of the link dynamics
    4. PID Joint control
    5. Model-based control

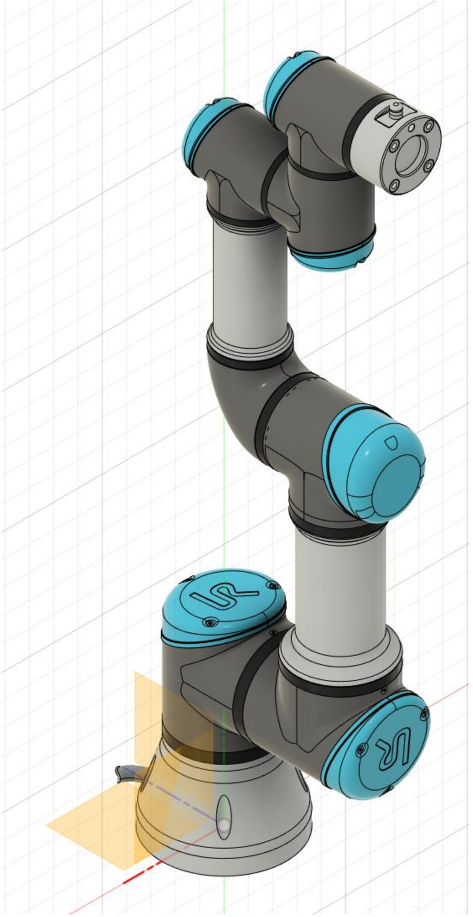


Single link, planar manipulator

Introduction

Why model planar link manipulators?

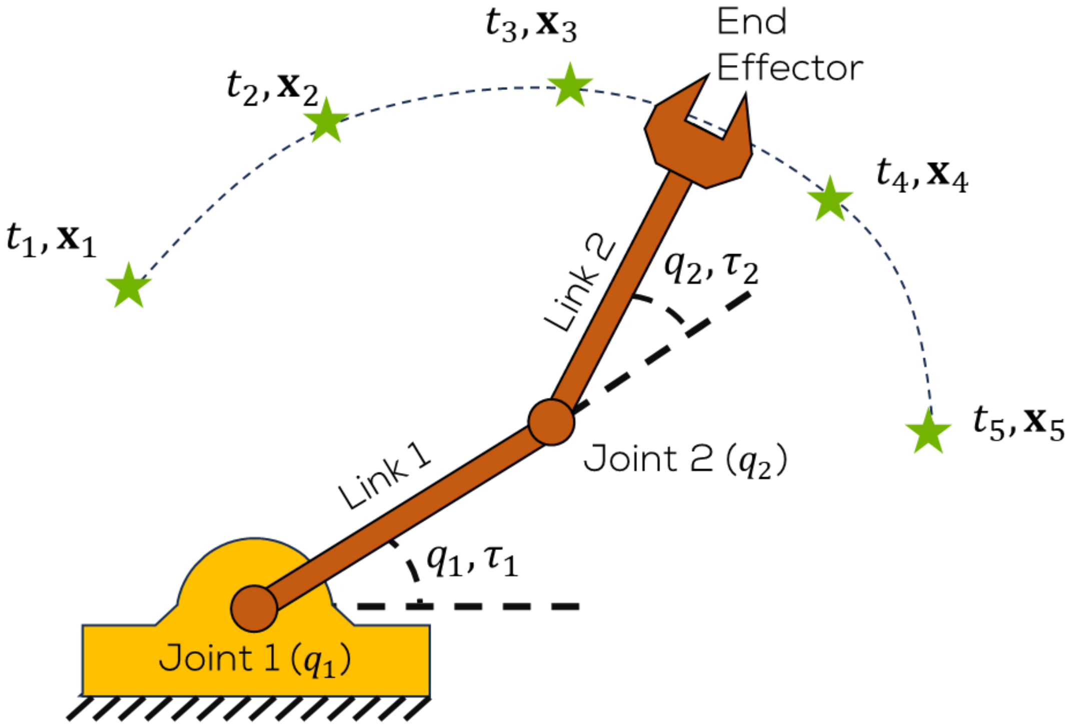
- Manipulators and Humanoid-type robots are like skeletons.
- They try to mimic the human form (or part of it) to perform different activities.
- As in the human body (mostly), each link rotates around a joint.
- Usually most of the joints have only one degree of freedom.
- The orientation of the joint determines the plane of movement for that link.#
- Each joint usually is driven by a motor.



Introduction

Manipulators

- A set of rigid, coupled links via joints.
- Manipulators are typically controlled by "actions" in each joints q_i; usually these actions come in the way of torques τ_i.
- Manipulators are affected by gravity and the load (payloads).
- Manipulator's end effector is typically commanded to follow a trajectory.



Double link, planar manipulator

SLM Manipulator

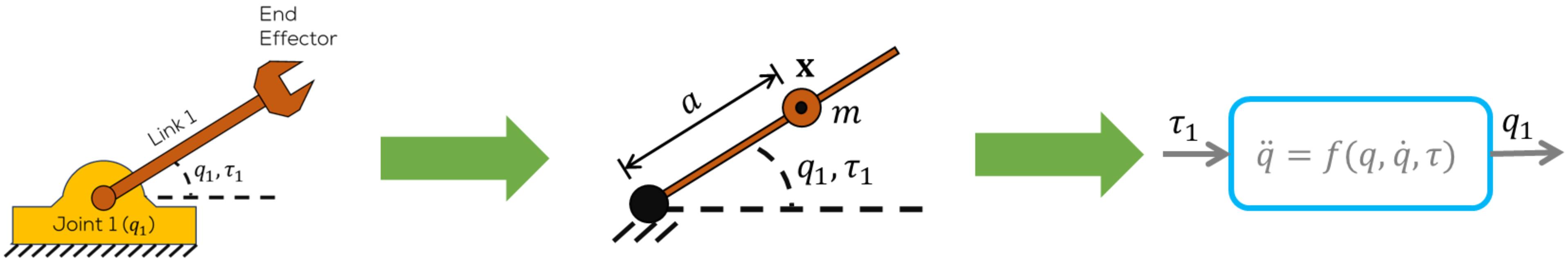
Dynamics

{Learn, Create, Innovate};

SLM Manipulator Dynamics

SLM Dynamics

- In the upcoming section, the derivation of the non-linear dynamics of an SLM will be presented.
- This analysis aims to enhance the comprehension of the intricate behaviour of the system and may be of interest to professionals working in this field.
- Several levels of abstraction will be performed.



Euler-Lagrange Mechanics

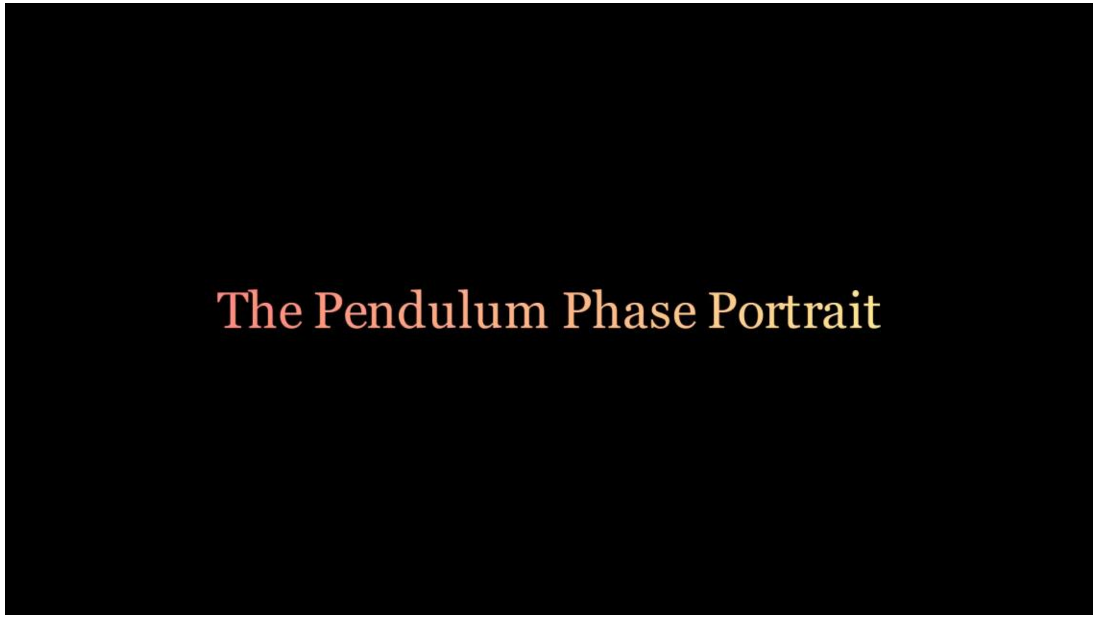
Introduction

- Dynamical systems are often described using Newton's Laws of motion.
- These laws state that by knowing the state of a system at a time t (initial conditions) and the forces acting on it, it is possible to know the evolution of the states by its governing equation (causality).
- This method is very useful for simple dynamical systems.
- Not so good for more complex systems e.g., Double pendulum.

Euler-Lagrange Mechanics

Introduction

- Every dynamical system describes a trajectory.
- The trajectory described by the system is given by the solution of the system's governing equation (differential equation).
- Deriving the governing equation of a system can be performed in multiple ways.
- Lagrange proposed a method using the total energy of a system



Euler-Lagrange Mechanics

Example (Inverse way)

- Let the equation of motion of a free-fall object be given by

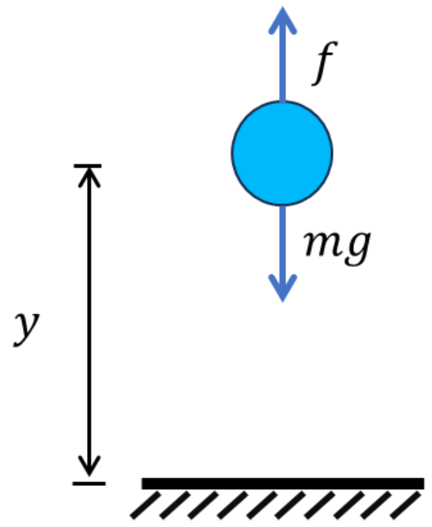
$$m\ddot{y} = f - mg$$

where m is the mass of the object, f f is the force acting on the object, and g is gravity.

- The left part of the equation can be substituted by

$$m\ddot{y} = \frac{d}{dt}(m\dot{y}) = \frac{d}{dt}\frac{\partial}{\partial\,\dot{y}}\left(\frac{1}{2}m\dot{y}^2\right) = \frac{d}{dt}\frac{\partial\kappa}{\partial\,\dot{y}}$$

where κ is the kinetic energy of the system



Euler-Lagrange Mechanics

Example (Inverse way)

- The right part of the equation can be substituted by

$$mg = \frac{\partial}{\partial y}(mgy) = \frac{\partial \mathcal{P}}{\partial y}$$

where $\mathcal{P}$ is the potential energy of the system

- Let $\mathcal{L}$ to be

$$\mathcal{L} = \kappa - \mathcal{P}$$

- Note that:

$$\frac{\partial \mathcal{L}}{\partial\, \dot{y}} = \frac{\partial \kappa}{\partial\, \dot{y}} \qquad and \qquad \frac{\partial \mathcal{L}}{\partial\, y} = -\frac{\partial \mathcal{P}}{\partial\, \dot{y}}$$

- Therefore, by substituting

$$m\ddot{y} = f - mg$$

$$m\ddot{y} + mg = f$$

$$\therefore \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial\, \dot{y}} - \frac{\partial \mathcal{L}}{\partial\, y} = f$$

- From the appropriate differential equation, the Euler LaGrange equation can be obtained.

Euler-Lagrange

For a **conservative, mechanical system**, the **Lagrangian** is the difference of the system's **kinetic** and **potential energies**:

$$L(q,\dot{q}) = KE(q,\dot{q}) - PE(q)$$

where $q, \dot{q}$ are the joint angle and joint angular velocity (states)

The Euler-Lagrange expression is used to derive the equation of motion from:

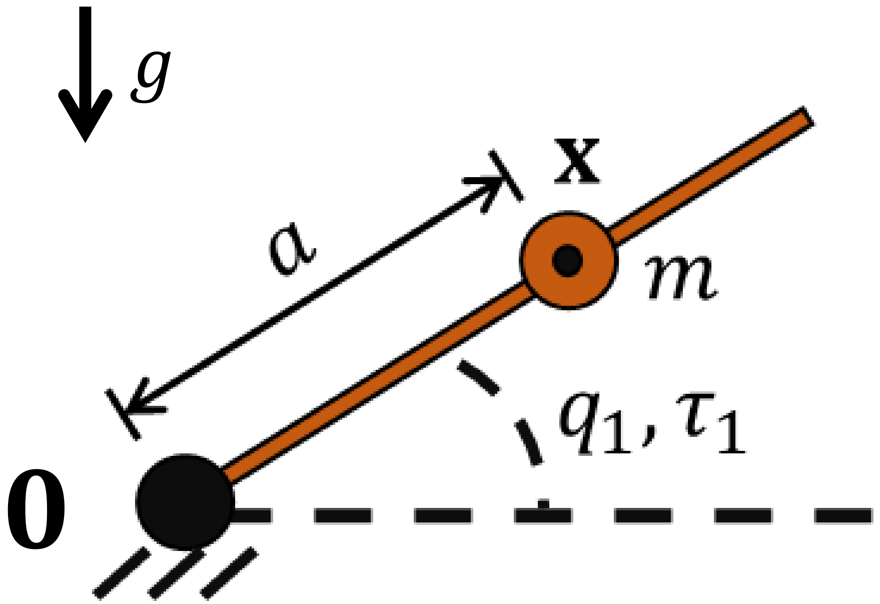
$$\frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial L(q,\dot{q})}{\partial q} = \tau$$

where τ is the applied joint torque.

- For manipulators, this gives a (mechanical) non-linear, 2nd order **equation of motion** of the form

$$\underbrace{m(q)\ddot{q}}_{\text{Mass – inertia "matrix"}} + \underbrace{c(q,\dot{q})\dot{q}}_{\text{Damping/ Coriolis Coefficient}} + \underbrace{g(q)}_{\text{Gravity "vector"}} = \underbrace{\tau}_{\text{Applied force "vector" along } q}$$

SLM Manipulator



- The single-link manipulator (SLM) consists of a rod connected to an actuator (usually a motor) that provides the input torque.
- The position and velocity of the centre of mass for the rod are

$$\mathbf{x} = a\begin{bmatrix}\cos(q)\\ \sin(q)\end{bmatrix},$$

$$\dot{\mathbf{x}} = a\begin{bmatrix}-\sin(q)\\ \cos(q)\end{bmatrix}\dot{q} \quad \Rightarrow \quad \|\dot{\mathbf{x}}\|^2 = a^2\dot{q}^2$$

**Because only considering angle above the axis*

SLM Manipulator

1. Express the PE & KE as functions of the states:

$$\mathbf{x} = a\begin{bmatrix}\cos(q)\\ \sin(q)\end{bmatrix}, \quad \|\dot{\mathbf{x}}\|^2 = a^2\dot{q}^2$$

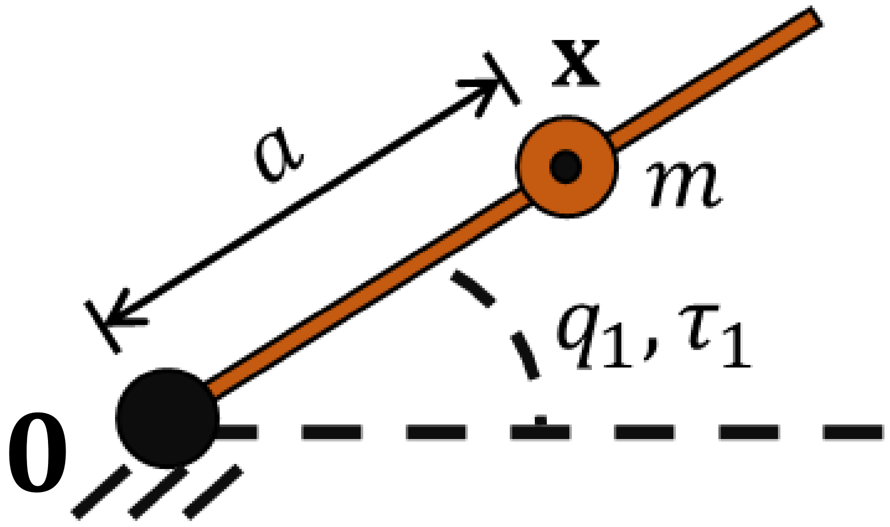
- Potential Energy (mgh) using the *forwards kinematics*:

$$PE = mgx_2 = mga\sin(q)$$

**Because only considering angle above the axis*

- Kinetic Energy using the *velocity kinematics:*

$$KE = \underbrace{\frac{1}{2}J\dot{q}^2}_{\text{Rotational KE}} + \underbrace{\frac{1}{2}m\|\dot{\mathbf{x}}\|^2}_{\text{Translational KE}} = \frac{1}{2}(J + ma^2)\dot{q}^2$$



SLM Manipulator

2. Derive the Lagrangian

$$L = KE - PE = \frac{1}{2}(J + ma^2)\dot{q}^2 - mga\sin(q)$$

3. Obtain the derivatives of the Euler Lagrange Equation

$$\frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial L(q,\dot{q})}{\partial q} = \tau$$

where:

$$\frac{\partial L(q,\dot{q})}{\partial q} = -mga\cos(q)$$

$$\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} = (J + ma^2)\dot{q} \;\; \Rightarrow$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) = (J + ma^2)\ddot{q}$$

4. Put it all together

$$\frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}}\right) - \frac{\partial L(q,\dot{q})}{\partial q} = \tau$$

$$\boxed{(J + ma^2)\ddot{q} + mga\cos(q) = \tau}$$

The **2nd order, non-linear equation of motion** for a single link manipulator, q, driven by an applied torque τ.

Non-linear SLM Equation of Motion

- The derived equation of motion for the single link manipulator can be related to the general equation:

$$(J + ma^2)\ddot{q} - mga\,cos(q) = \tau$$

$$\underbrace{m(q)\ddot{q}}_{\text{Mass – inertia “matrix”}} + \underbrace{c(q,\dot{q})\dot{q}}_{\text{Damping/ Coriolis Coefficient}} + \underbrace{g(q)}_{\text{Gravity “vector”}} = \underbrace{\tau}_{\text{Applied force “vector”}}$$

- The ODE's **order is 2** (dimension of the state space: q, $\dot{q}$)
- The **torque**, τ, is the **input** to the mechanical system

- The dynamics are **non-linear** because of cos(q) (gravity "vector")
- There is **no damping** ($\dot{q}$ term) so we may expect **pendulum**-type natural/unforced dynamics which do not decay (conservative system or no friction assumption)
- The mass/inertia "matrix" is constant. Note that the expression for the link's inertia is very similar to the other COM term.

Summary

$$(J + ma^2)\ddot{q} + mga\cos(q) = \tau$$

We used **Euler-Lagrange** to derive the mechanical, non-linear equation of motion of a planar, single link manipulator

- For a single link, this is a bit overkill and the equation of motion could have been derived from a torque balance perspective
- However, it is necessary to use a principled approach to derive the dynamics of manipulators with 2 or more links (next week)
- The equations are **2^{nd} order** (angular force -> angular acceleration) and the **non-linearity** arises from the cos(q) gravity term
- The (unforced) dynamics are that of a **pendulum**
- The dynamics can easily be simulated in Matlab, Simulink, MuPAD

Understanding the Equation of Motion

Torque used to counter gravity (PE)

$$(J + ma^2)\ddot{q} + \boxed{mga\,cos(q)} = \tau$$

- This is the joint torque due to gravity acting through the centre of mass,
- Maximum when horizontal, $q = 0, \pi, \ldots$

Torque required to accelerate the link (KE)

$$\boxed{(J + ma^2)\ddot{q}} + mga\,cos(q) = \tau$$

- Joint torque used to accelerate the link. Newton's 2nd law of rotation, $\tau = J'\ddot{q}$
- For a uniform link, $J' = (4/3)ma^2$
- This is just the inertia of rod rotating around end point (not it's COM, parallel axis theorem)

Each additive term can be regarded as a torque and the overall equation represents a balance of applied & resulting torques

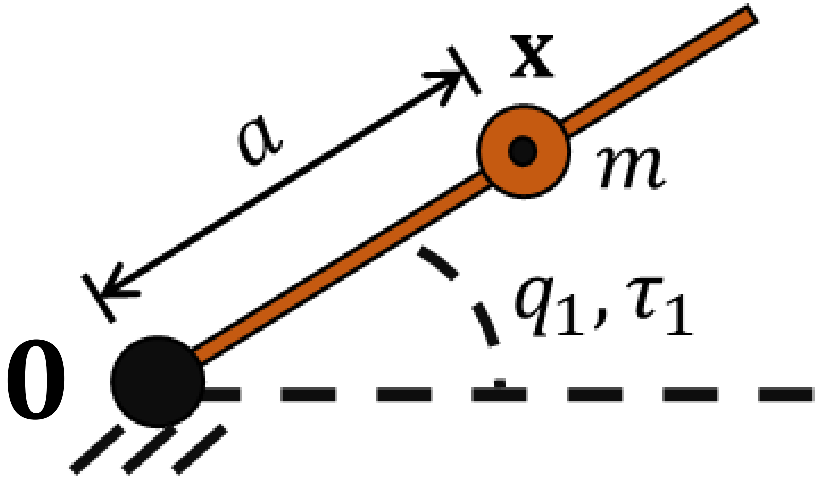
SLM Manipulator

- Let the following parameters be used for the SLM. (These parameters will be used in this ppt, unless otherwise specified)

$$m = 3\ kg,$$
$$g = 9.8ms^{-2}$$
$$a = 0.2m$$
$$J = \frac{1}{3}ma^2 = 0.04\ kgm^2$$

- Using the previous parameters, produces the non-linear equation of motion

$$\ddot{q} + 36.75\cos(q) = 6.25\tau$$
$$q(0) = 1\ rad$$
$$\dot{q}(0) = 0\ rad/s$$

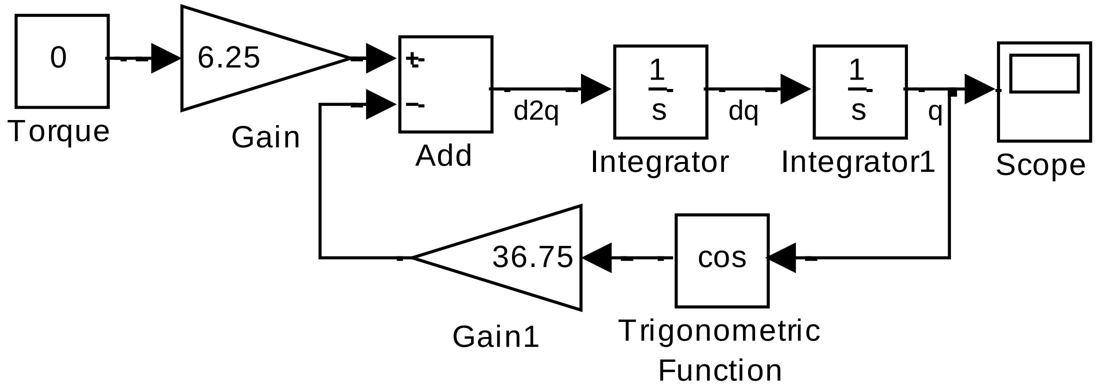


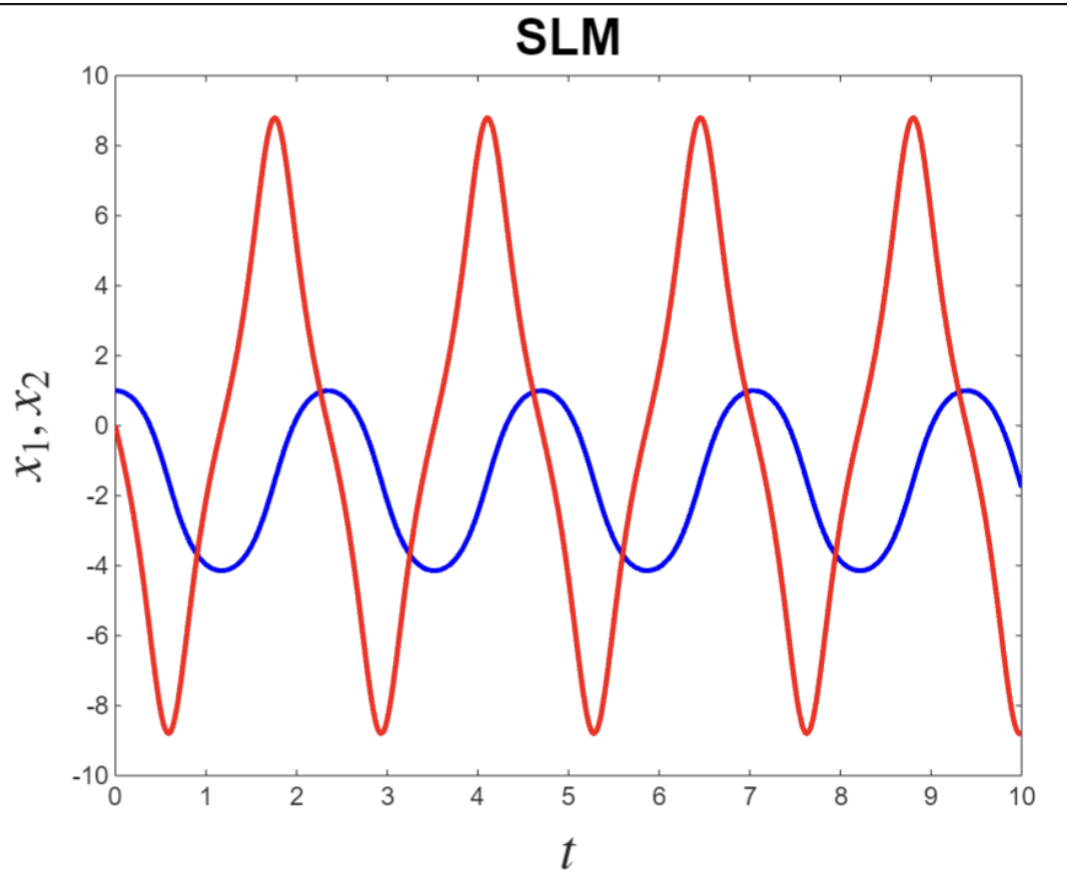
SLM Manipulator

Unforced response

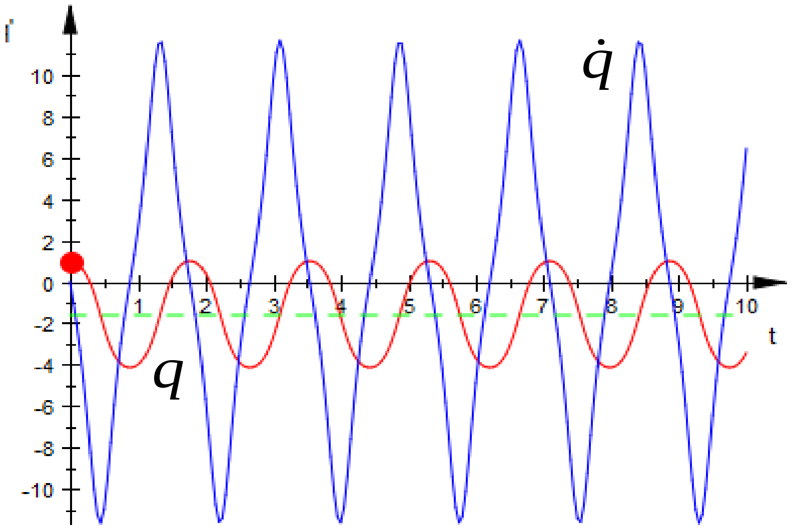
- Unforced response (autonomous response), where the systems uses its own energy (does not depend on time)

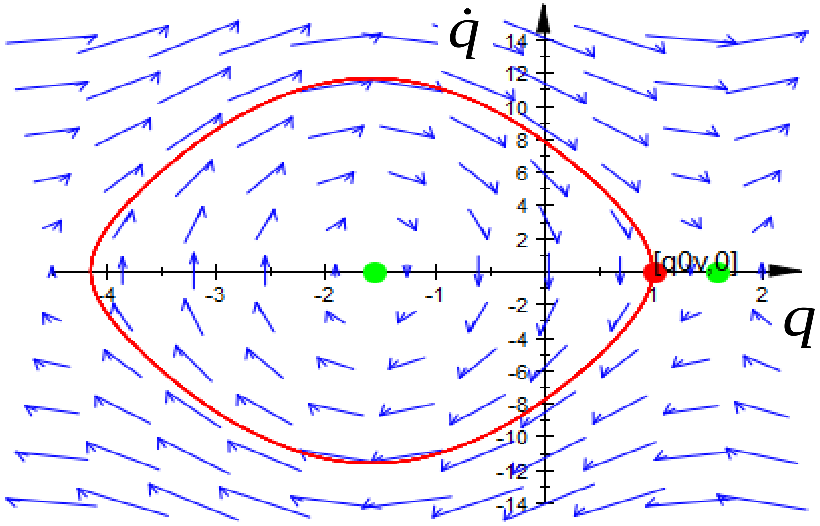
$$\tau(t) = 0Nm$$





SLM Manipulator





- For the initial point [1,0], the pendulum oscillates between $q = 1$ & $q = -4.141$ (rad) with a time period of approximately 1.8 s.
- The periodic motion is approximately sinusoidal (but squashed)
- There are two equilibrium points (green) for the vertical positions
- The general behaviour is that the pendulum either oscillates about the downwards vertical, -π/2, or, for a large enough initial velocity, it keeps rotating in the same direction

Analysis of SLM Dynamics

- We've obtained the non-linear equation of motion and simulated the unforced, oscillatory response
- A local, linear analysis of the dynamic equation should reveal interesting dynamic properties associated with the system.

$$\tau \rightarrow \boxed{\ddot{q} = f(q, \dot{q}, \tau)} \rightarrow q$$

- Poles (time constants, frequency)
- Gain
- Relate to physical parameters

1. Torque balance (equilibrium) analysis
2. Plant linearization (state space form)
3. Linearized, non-linear comparative simulation
4. Poles of the linearized system (unstable, oscillatory, …)
5. Gain of the linearized system

Torque Equilibrium (Balance) Analysis

To hold the link in **equilibrium**: $\dot{q} = 0, \quad \ddot{q} = 0$

$$(J + ma^2)\ddot{q} + mga\cos(q) = \tau$$

the applied torque is

$$\tau = mga\cos(q)$$

This is **maximum** when the arm is horizontal $(q = 0, \pi, \ldots)$ as gravity has the largest effect

$$\tau = \pm mga$$

- For the default manipulator parameters, the actuator (motor & gearbox) must be capable of delivering a torque of

$$|\tau| \geq 5.89Nm$$

- To drive the manipulator, the actual torque must be greater than this value & the resulting angular acceleration is inversely proportional to the inertia.

Example: Torque Balance

The 2nd order ODE of the exemplar manipulator is:

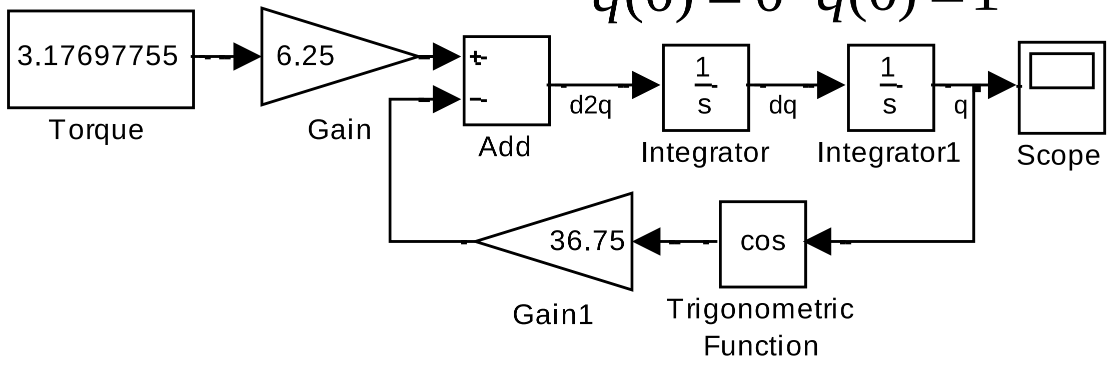
$$\ddot{q} + 36.75\cos(q) = 6.25\tau$$

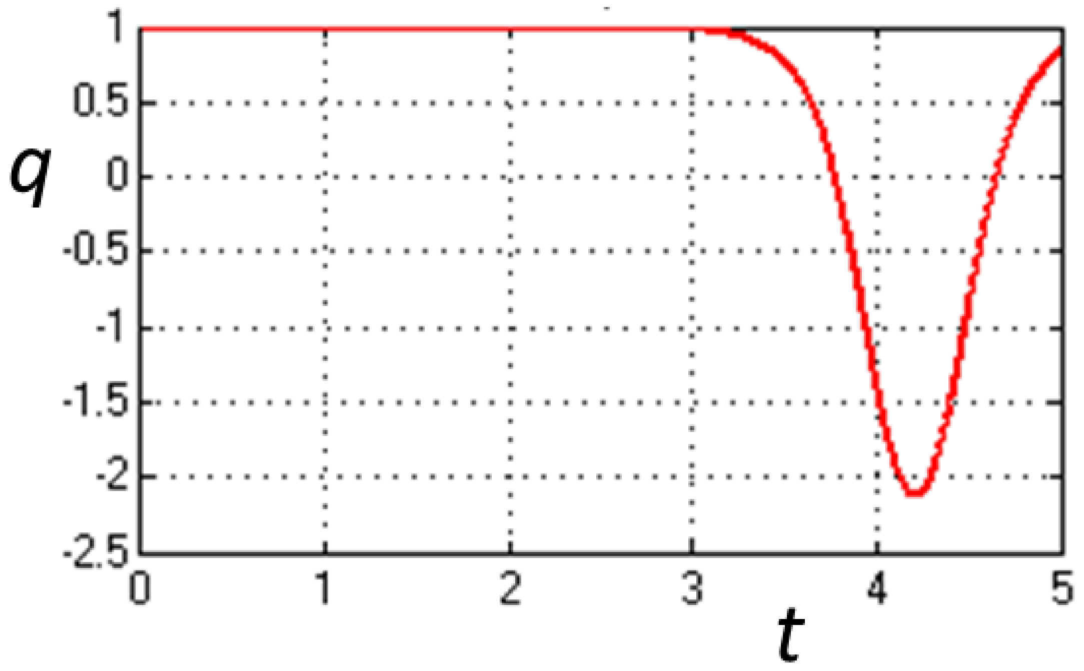
and, at q=1 rad, the **equilibrium torque** is given by (to 8 d.p.)

$$\tau = mga\cos(q) = 3.17697755\ Nm$$

Simulating in Simulink

$$\dot{q}(0) = 0 \quad q(0) = 1$$





Does the system stay at the equilibrium point?

We'll see that this equilibrium is unstable

Linearisation

- We'd like to analyse and understand the **local** behaviour of the manipulator about a (steady state) operating point $\{q^*, \tau^*\}$
- Use a 1st order, multivariate Taylor series for

$$\ddot{q} = f(q, \tau)$$

$$\ddot{q} = (\tau - mga\cos(q))/(J + ma^2)$$

$$\boxed{\ddot{q}(q^* + \Delta q, \tau^* + \Delta\tau) = \ddot{q}(q^*, \tau^*) + \frac{\partial f(q^*, \tau^*)}{\partial q}\Delta q + \frac{\partial f(q^*, \tau^*)}{\partial \tau}\Delta\tau}$$

Multivariate Taylor Expansion

$$\boxed{\Delta\ddot{q} = \frac{\partial f(q^*, \tau^*)}{\partial q}\Delta q + \frac{\partial f(q^*, \tau^*)}{\partial \tau}\Delta\tau}$$

$$\Delta\ddot{q} - \frac{mga\sin(q^*)}{(J + ma^2)}\Delta q = \frac{1}{(J + ma^2)}\Delta\tau$$

This produces a 2nd order, linear ODE in the signals

$$\Delta q = q - q^*, \quad \Delta\tau = \tau - \tau^*$$

The linearized Δq term represents the "gravity component"

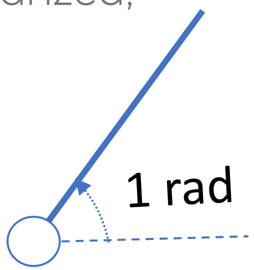
Linearised System

$$\Delta\ddot{q} - \frac{mga\sin(q^*)}{(J+ma^2)}\Delta q = \frac{1}{(J+ma^2)}\Delta\tau$$

- Linearizing about $q^* = 1$ rad, the linearized, exemplar, 2nd order ODE is

$$\Delta\ddot{q} - 31\Delta q = 6.25\Delta\tau$$

$$\ddot{y} - 31y = 6.25u$$



The input and output signals are **increments** about the set point:

$$y \triangleq \Delta q = q - q^*, \quad u \triangleq \Delta\tau = \tau - \tau^*$$

- The **steady state torque** is given by (to 2 decimal places)

$$\tau^* = mga\cos(q^*) = mga\cos(1) = 3.18$$

- The 1st order, truncated Taylor series is only valid in a **local region**, where 2nd and higher derivatives can be neglected
- We'd expect this approximation to be valid within approximately ±0.5 rad about the linearization point q^*, (approximation of a sinusoid)

Example: Non-linear / Linearized Simulation

- To compare the local dynamical behaviour of the non-linear and the linearized dynamic models, consider linearizing the exemplar system about $\{q^*, \tau^*\} = \{1, 3.18\}$ and considering the **unforced** (autonomous) response ($\tau = 0$)

- Let both systems (linear and nonlinear) to be started from the same point (initial conditions) $\{q(0), \dot{q}(0)\} = \{0.8, 0\}$ which is close to the linearization point

1. Non-linear simulation

$$\ddot{q} = 6.25\tau - 36.75\cos(q)$$

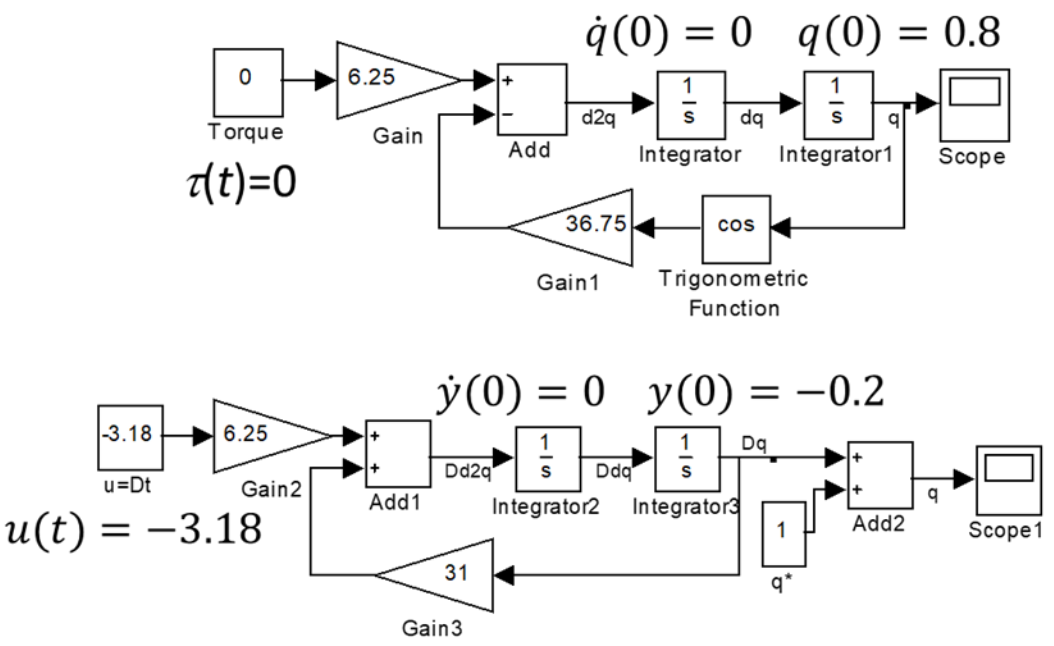
2. Linearized simulation

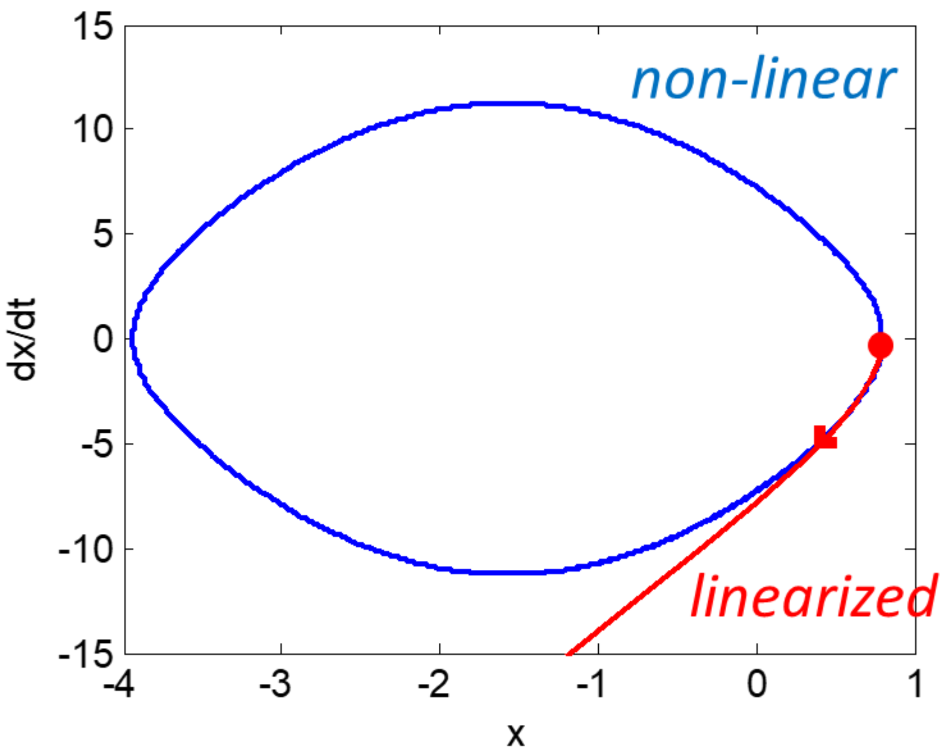
$$\ddot{y} - 31y = 6.25u$$

$$y \triangleq \Delta q = q - q^*$$

$$u \triangleq \Delta\tau = \tau - \tau^*$$

Example: Non-linear / Linearized Simulation





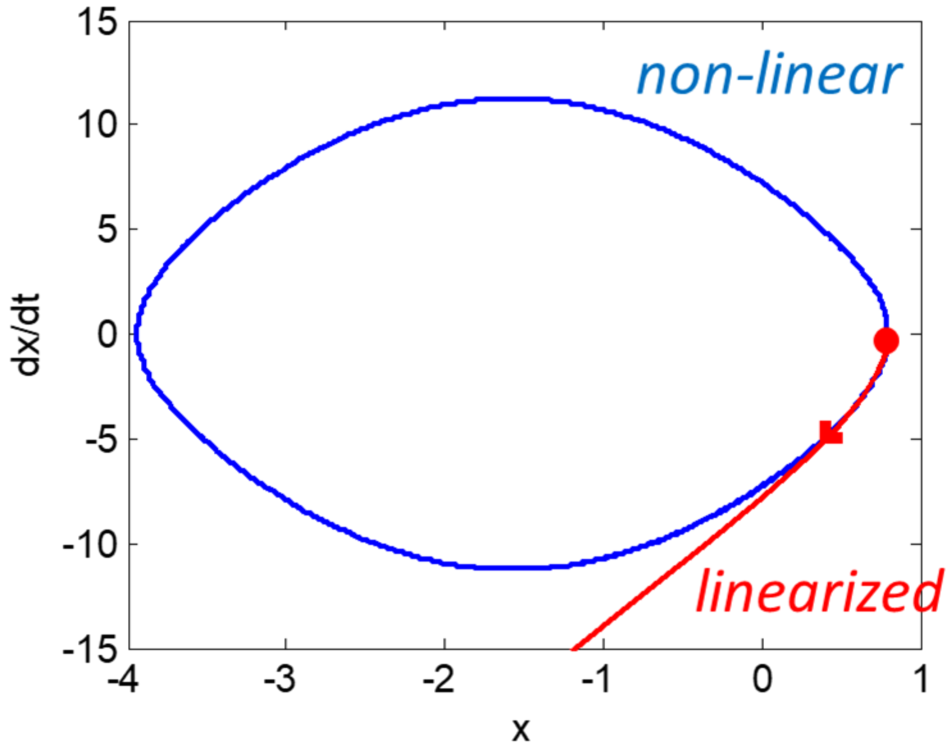
Example: Non-linear / Linearized Simulation

1. Non-linear (unforced) simulation (blue)

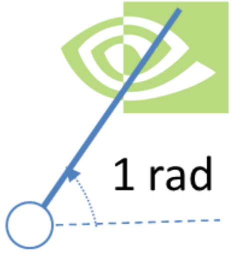
- Near sinusoidal, periodic orbit of a pendulum which starts from $q(0) = 0.8, \dot{q}(0) = 0$ and therefore oscillates between 0.8 and -3.94 rad

2. Linearized simulation (red)

- Initial close agreement with the non-linear model
- After ~0.5 rad, the linearized response begins to diverge (why?)
- The unstable, linearized response $\{q, \dot{q}\} \rightarrow -\infty$ and is certainly not periodic



Linear SLM Analysis (Poles) $q^* = 1$



1. The linearized (about $q^* = 1$), exemplar system is:

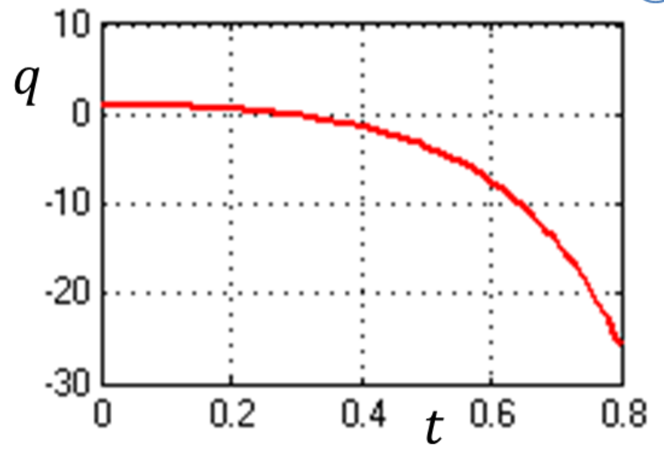
$$\ddot{y} - 31y = 6.25u$$

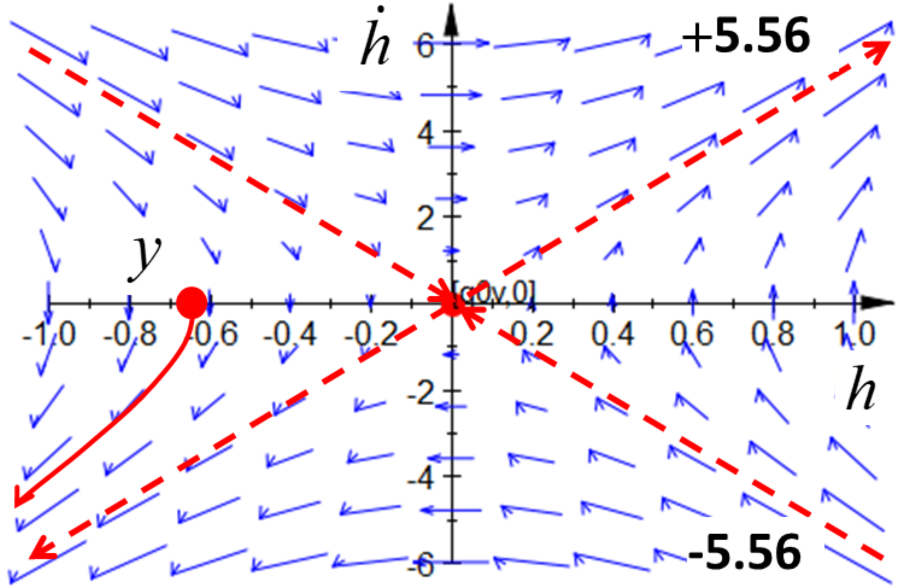
Poles $s^2 - 31 = 0$

Poles & unforced response $h(t)$

$$s = \sqrt{31} = \pm 5.56$$

$$h = Ae^{5.56t} + Be^{-5.56t}$$

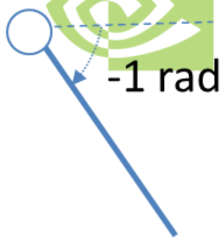




Pole analysis:

- Unforced, h, unstable behaviour (blue arrows)
- Time constant 0.18 s = 1/5.56
- Unstable mode in positive & negative quadrants (red dashes)
- $\Delta\tau$ causes a "step u" response (red solid line)

Linear SLM Analysis (Poles) $q^* = -1$



Linearizing (Taylor series) about $q^* = -1$ rad, the exemplar, locally linear state space system is:

$$\ddot{y} + 31y = 6.25u$$

$$s^2 + 31 = 0$$

Poles & unforced response $h(t)$

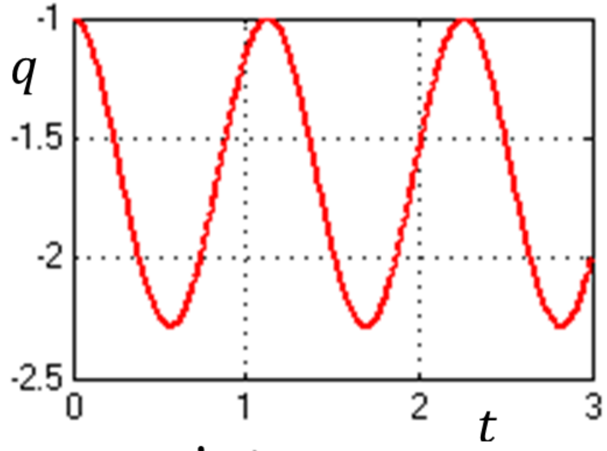
$$s = \sqrt{-31} = \pm 5.56j$$

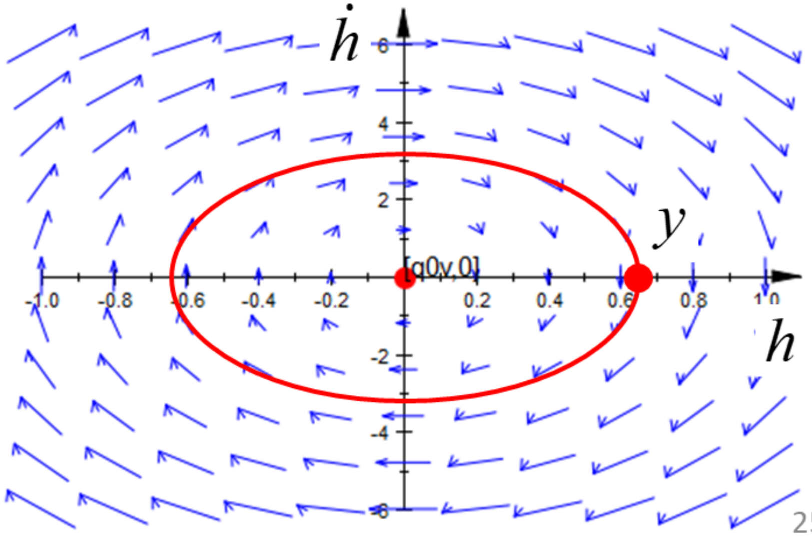
$$h = Ae^{j5.56t} + Be^{-j5.56t}$$

$$= C\cos(5.56t - D)$$

Pole analysis

- Unforced, h, local sinusoidal behaviour (blue arrows)
- Time period T=1.13 s
- $\Delta\tau$ causes a "step u" response (red solid line), not symmetric about $-\pi/2$ as q: 1->2.284





General Expression for SLM Poles

- Linearizing (Taylor series) about q^*, the locally linear ODE is:

$$\Delta\ddot{q} - \frac{mga\sin(q^*)}{(J + ma^2)}\Delta q = \frac{1}{(J + ma^2)}\Delta\tau$$

- The **poles** (exponents) are given by $s = \pm\sqrt{mga\sin(q^*)/(J + ma^2)}$

Behaviour is unstable or oscillatory depending on the sign of sin(q^*)

1. **Above horizontal,** $0 < q^* < \pi$, so $0 < \sin(q^*) \leq 1$

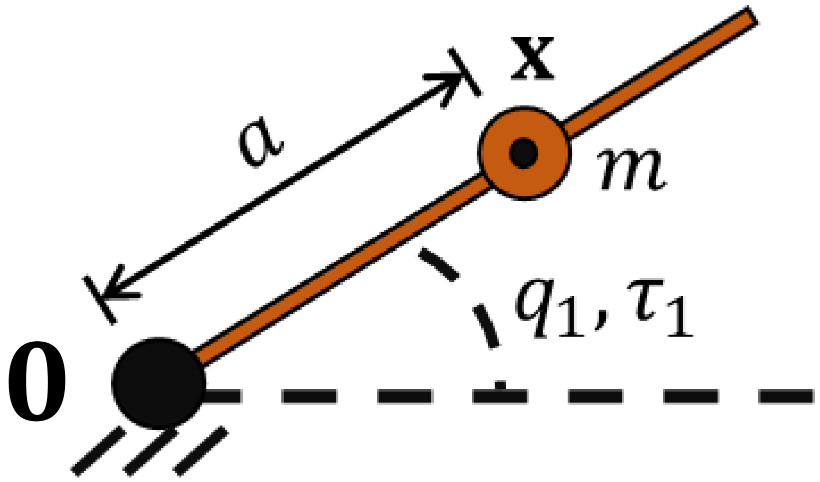
   Real poles, unstable, $s = \pm\sqrt{mga\sin(q^*)/(J + ma^2)}$

2. **Horizontal**, $q^* = \{0, p, \ldots\}$, so $\sin(q^*) = 0$

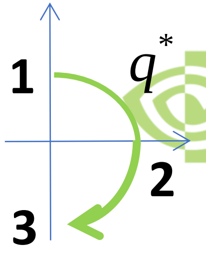
   Two poles which are *both zero*, a *double integrator* (freefall)

3. **Below horizontal**, $-\pi < q^* < 0$, so $-1 \leq \sin(q^*) < 0$

   Imaginary poles, oscillatory, $s = \pm j\sqrt{-mga\sin(q^*)/(J + ma^2)}$

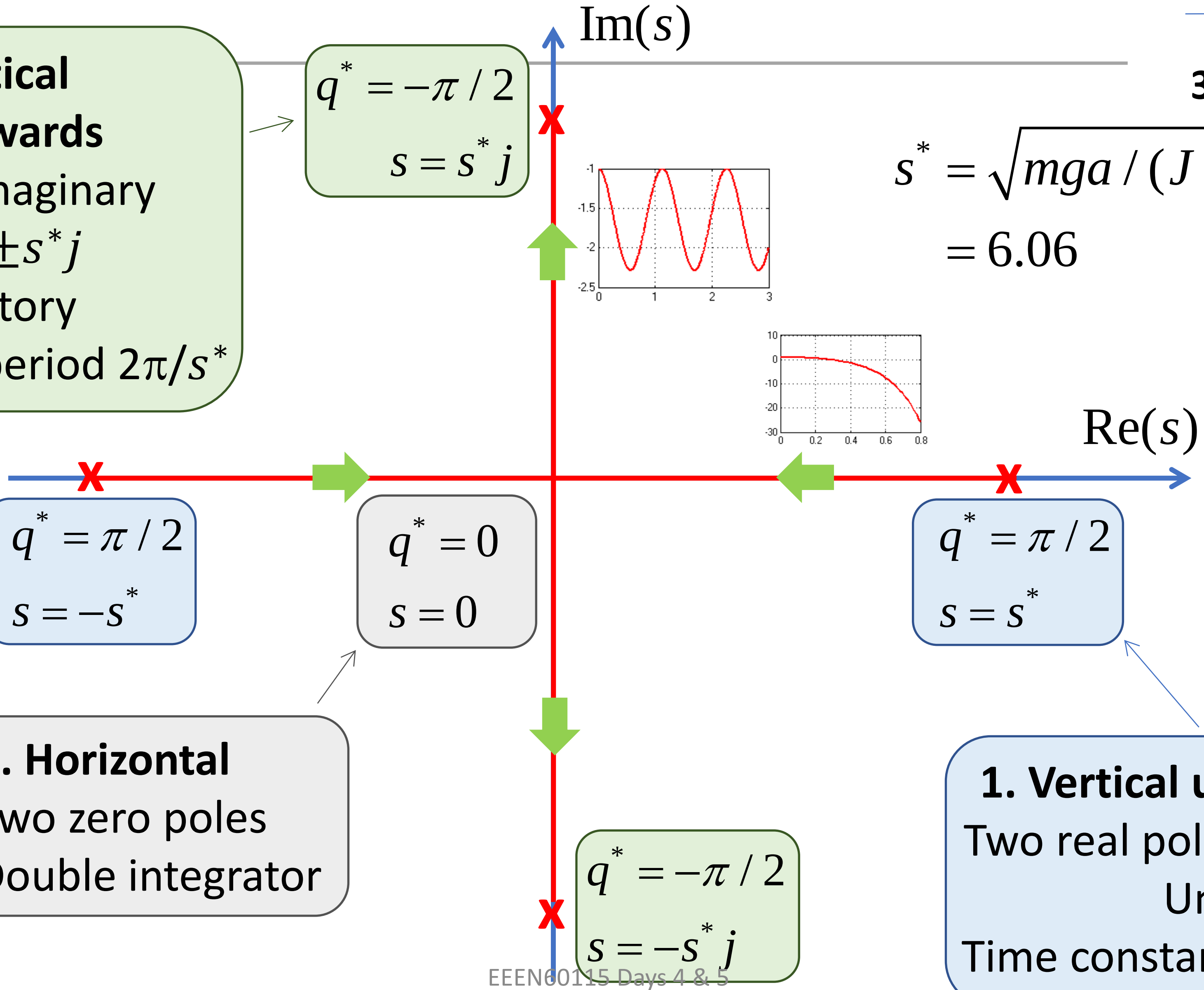


3.4 Pole Locus Plot: Varying q^*



$$s^* = \sqrt{mga / (J + ma^2)}$$
$$= 6.06$$

3. Vertical downwards
Two imaginary poles $\pm s^* j$
Oscillatory
Time period $2\pi/s^*$



2. Horizontal
Two zero poles
Double integrator

1. Vertical upright
Two real poles $\pm s^*$
Unstable
Time constant $1/s^*$

Pole Relationship to SLM Parameters

Let's focus on a **uniform link** so $J = ma^2/3$. The interpretation for other link shapes is similar.

The inverse **time constants** (real poles) and the **angular frequency** (imaginary poles) are bounded by

$$\boxed{s^* = \sqrt{3g/4a}}$$

independent of mass

1. *Link standing vertically up,* $s = \pm\sqrt{3g/4a}, \quad \tau = \sqrt{4a/3g}\ s$

- **Increasing** the **link length** increases the **time constant** (sqrt)
- **Increasing gravity** increases the **time constant** (sqrt)

2. *Link hanging vertically down,* $\omega = \pm\sqrt{3g/4a}, \quad T = 4\pi\sqrt{a/3g}\ s$

- **Increasing** the **link length** increases the **time period** (sqrt)
- **Increasing gravity** decreases the **time period** (sqrt)

Other linearization points are similar because of the $\sin(q^*)$ factor

Gain of the Linearized SLM

Linearizing (Taylor series) about q^*, the linear ODE is:

$$\Delta\ddot{q} - \frac{mga\sin(q^*)}{(J + ma^2)}\Delta q = \frac{1}{(J + ma^2)}\Delta\tau$$

The steady state or DC **gain** is: $\frac{\Delta q}{\Delta\tau} = -\frac{1}{mga\sin(q^*)}$

The gain's sign depends on the position of the link

1. **Above horizontal**, $0 < q^* < \pi$, so $0 < \sin(q^*) \leq 1$

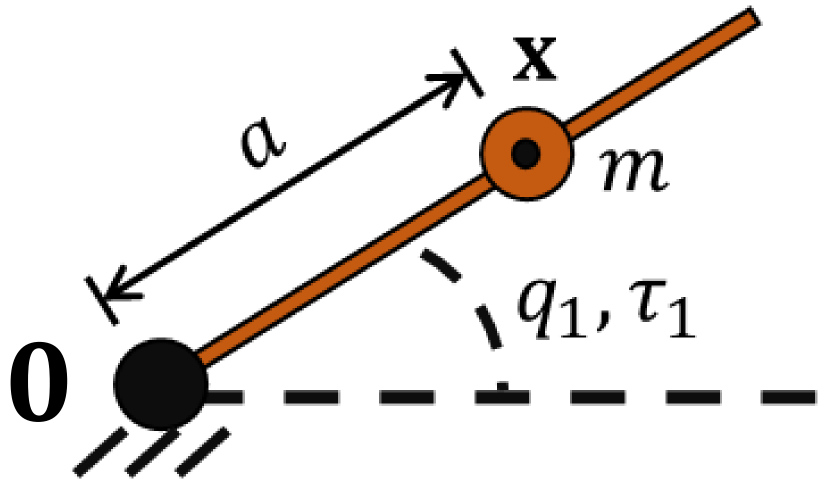
   Gain is negative, in steady state increasing τ^* decreases q^*

2. **Horizontal**, $q^* = \{0, \pi, \dots\}$, so $\sin(q^*) = 0$

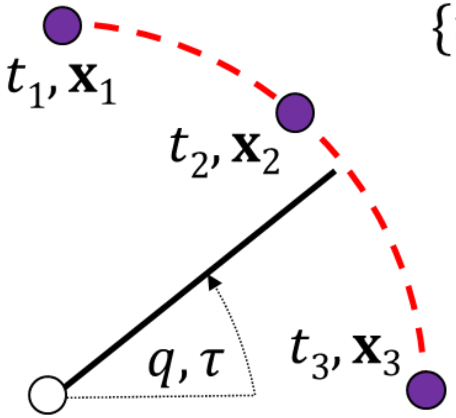
   Gain is not defined (infinite) as the system is a double integrator

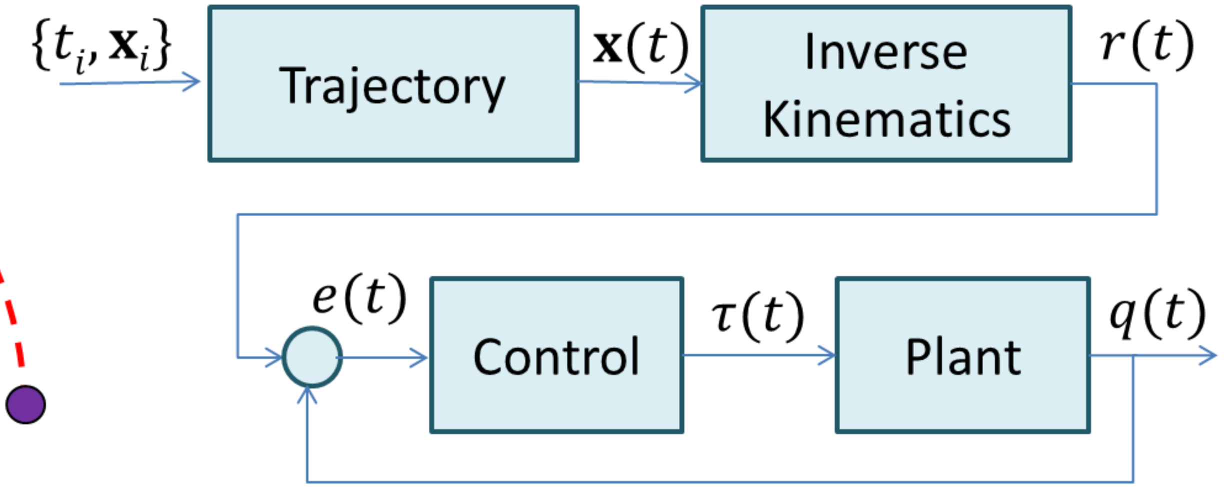
3. **Below horizontal**, $-\pi < q^* < 0$, so $-1 \leq \sin(q^*) < 0$

   Gain is positive, in steady state increasing τ^* increases q^*



SLM PID Joint Control





- In this section a controller for the SLM will be developed.
- The controller is based on the classic PID control strategy.
- We'll now consider developing a position (set-point or tracking) controller for a planar (vertical), single link manipulator (without motor dynamics)

Why PID Control?

Feedback Control

- Feedback control deals with designing an error-based controller to achieve performance (accuracy, time), robustness & energy objectives.

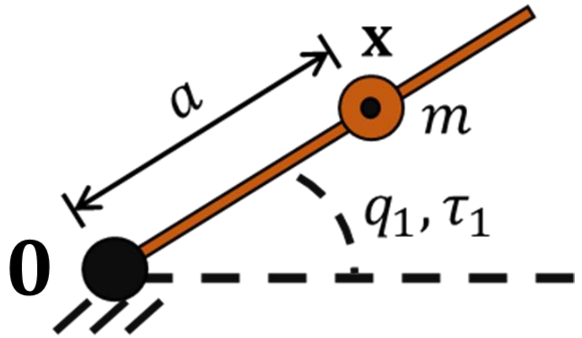
Feedback control is used to

- Improve transient response time
- Track reference set points as they're error based.
- Improve robustness to parameter variation
- Balance expended energy to performance / robustness objectives

PID feedback control is widely used and will be applied to / analysed for the joint control problem

Usual PID assumption that the **reference joint angle** is specified as a **step** sequence (not smooth splines)

$$(J + ma^2)\ddot{q} + mga\cos(q) = \tau$$



SLM & PID Joint Control

Plant

- Plant is 2nd order, **non-linear**

$$(J + ma^2)\ddot{q} + mga\cos(q) = \tau$$

- **Linearize** about the point $\{\tau^*, q^*\}$

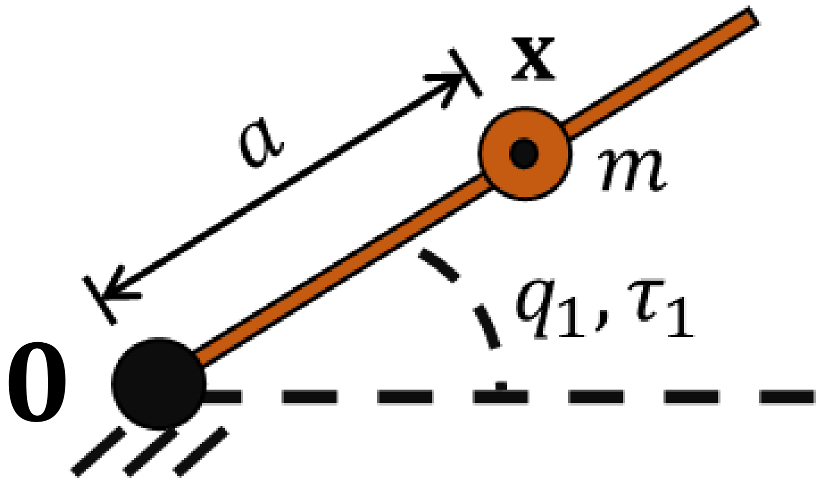
$$\Delta\ddot{q} - \frac{mga\sin(q^*)}{(J + ma^2)}\Delta q = \frac{1}{(J + ma^2)}\Delta\tau$$

- This is equivalent to:

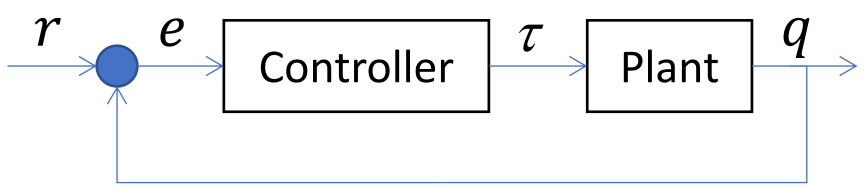
$$\ddot{y} + a(q^*)y = bu$$

* Linearized 2nd order system has **no damping** and dynamics are either:

- **Unstable** (above horizontal)
- **Oscillatory** (below horizontal)



SLM & PID Joint Control



- Propose a **fixed PID controller**

$$u = k_P e + k_I \int e + k_D \dot{e}$$

- Substitute it in the following equation

$$\ddot{y} + a(q^*)y = bu$$

- Therefore

$$\dddot{y} + bk_D\ddot{y} + (a(q^*) + bk_P)\dot{y} + bk_I = b(k_P\dot{r} + k_I r + k_D\ddot{r})$$

- D term for **damping**
- I term for **set point tracking**
- 3rd order, **2 additional zeros**
- Can't directly apply pole-zero cancellation because of the **unstable pole**
- P term must be large enough (**robust**) to ensure system is **stable, i.e.,** $(a(q^*) + bk_P) > 0$

$$k_P > -a(q^*)/b$$

Derivative Error or State Feedback?

- Often, when the reference signal is given as a (sequence of) steps, the derivative part of the control signal is simplified to:

$$u = -k_D\dot{y}$$

rather than

$$u = k_D\dot{e}$$

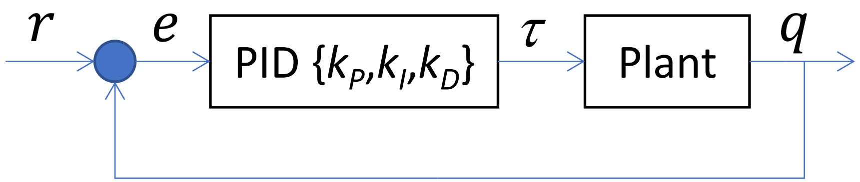
- In the **s-domain**, this has the effect of **removing** one of the **zeros** (right hand side, r) while still adding the same "damping" term to the transient dynamics (left hand side, y), hence the poles are the same.

- In the time-domain, the two signals are identical apart from the time at which the step occurs. Differentiating a step produces an impulse (zero everywhere apart from at t=0, where the value is infinite). This would produce a spike in the (demanded) torque and a corresponding discontinuity in the joint acceleration.

- See the low pass filter derivative term the PID block in Simulink

Introduction to PID Pole Placement

There are many different methods to design the PID controller's parameters.



We're going to approach the problem by performing **pole placement** (similar to state space) and then analysing the closed loop performance to determine if the

- **zeros** (determine overshoot, non-minimum phase behaviour)
- **robustness** (performance & stability for parameter variation)
- demanded **torque** (parameters not too large) are acceptable.

Pole Placement Calculation

When talking about **poles**, we're referring to:

- The values of s which makes the denominator of the transfer function go to zero.
- The values of s (assuming $y \sim e^{st}$) in the auxiliary equation when using complementary functions.
- The (complex: **real** and **imaginary** components) **exponents** in the exponentials which **scale time** (**time constant** and **time period**) and hence describe the system's **impulse response** (transient)

They're equivalent. We're talking about exponential signals

$$y \sim e^{st} = e^{(\sigma + j\omega)t} = e^{\sigma t}(\cos(\omega t) + j\sin(\omega t))$$

By specifying the poles, we specify the **time constants** and **time periods** (when the poles are complex valued).

This does **not** completely specify the closed loop response (initial values, amplitudes, …), but does determine how **fast** the transients are (scale the time axis)

Pole Placement Specification

- For a robotics application, an approximate **transient time of 1 s** with **no oscillations** would be "good".
- This means that the time constant(s) should have approximately 0.2 s. Therefore, the pole(s) should be **real valued** and located at **s=-5**.

$$\dddot{y} + bk_D\ddot{y} + (a(q^*) + bk_P)\dot{y} + bk_I = b(k_P\dot{r} + k_I r + k_D\ddot{r})$$

$$H(s) = \frac{bk_P(s + k_I/k_P)}{s^3 + bk_D s^2 + (a(q^*) + bk_P)s + bk_I}$$

- For a PID joint controller, there are **3 poles** and also **3 controller parameters**!
- As long as the $s = -5$ is the slowest pole, the other two can be located at different points, $s < -5$ (further to the left on the real axis).
- Having **repeated poles** ($s = -5$ three times) ensures:
  - a critically damped response / no overshoot (assuming no zero(s))
  - minimize the control energy, as faster (larger, negative) poles requires greater control energy

Pole Placement Calculation

- Calculate the auxiliary equation

$$(s+5)(s+5)(s+5)=0$$

$$s^3+15s^2+75s+125=0$$

- Using this auxiliary equation we can determine the controller parameters as follows

$$s^3+bk_Ds^2+(a(q^*)+bk_P)s+bk_I=0$$

Therefore

$$s^2: \quad k_D=15/b$$

$$s: \quad k_P=(75-a(q^*))/b$$

$$1: \quad k_I=125/b$$

- Note that when the desired time constants are small (< 1 s), the coefficients of the desired auxiliary equation are increasing in size and hence there is generally a **natural ordering** of the **PID coefficients**

$$k_D<k_P<k_I$$

Pole Placement Calculation

- The PID controlled, closed loop transfer functions for the generic and the exemplar systems are given by:

$$H(s) = \frac{bk_P(s + k_I/k_P)}{s^3 + bk_D s^2 + (a(q^*) + bk_P)s + bk_I}$$

Therefore

$$H(s)$$

$$= \frac{6.25k_P(s + k_I/k_P)}{s^3 + 6.25k_D s^2 + ([-36.75, 36.75] + 6.25k_P)s + 6.25k_I}$$

- Remember

$$k_P > \max(-a(q^*)/b, 0)$$

$$k_P > 5.88$$

We require the desired closed loop to have (three) repeated poles at $s = -5$, which ensures no overshoot (assuming no zeros) and a transient duration of ~ 1 *s*.

Take the controller design point at $q^* = 0$ (the horizontal). This can be weakly justified as being in the mid-point of the $a(q^*)$ coefficient variation as $a(0) = 0$.

Then solving the pole placement problem gives:

$$k_P = 12, \quad k_I = 20, \quad k_D = 2.4$$

Time Constant Robustness Analysis 1

- We have assumed that the desired response has three repeated poles at $s = -5$ and that the system was linearized at the horizontal or $q^* = 0$

$$k_P = 12, \quad k_I = 20, \quad k_D = 2.4$$

- For other linearization points, the system will be stable as $k_P > 5.88$, but the transient response will not be three repeated poles.

- Taking the two vertical positions as extreme values:

Vertical upright: $q^* = \pi/2$,

$$s^3 + 15s^2 + 38.25s + 125 = 0$$

$$s = \{-12.77, -1.11 \pm 2.92j\}$$

Vertical down: $q^* = -\pi/2$,

$$s^3 + 15s^2 + 111.75s + 125 = 0$$

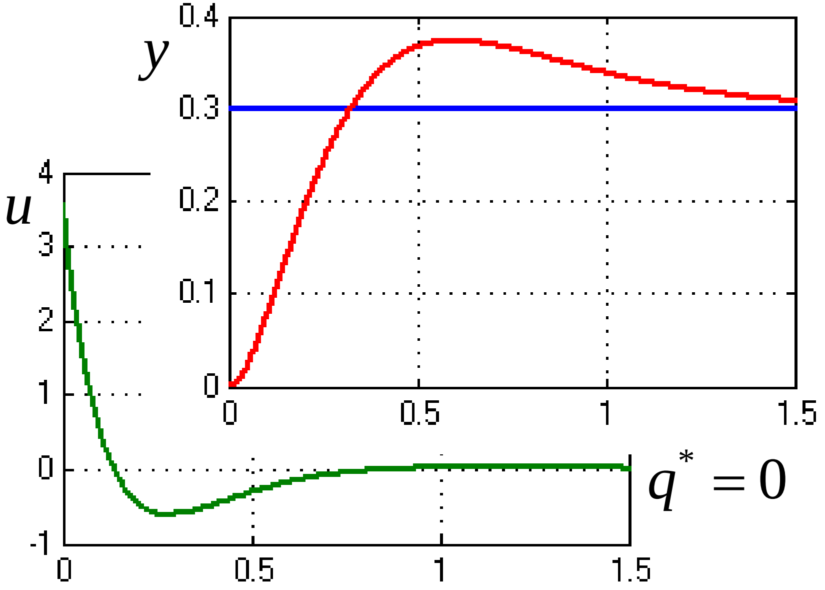
$$s = \{-1.33, -6.83 \pm 6.84j\}$$

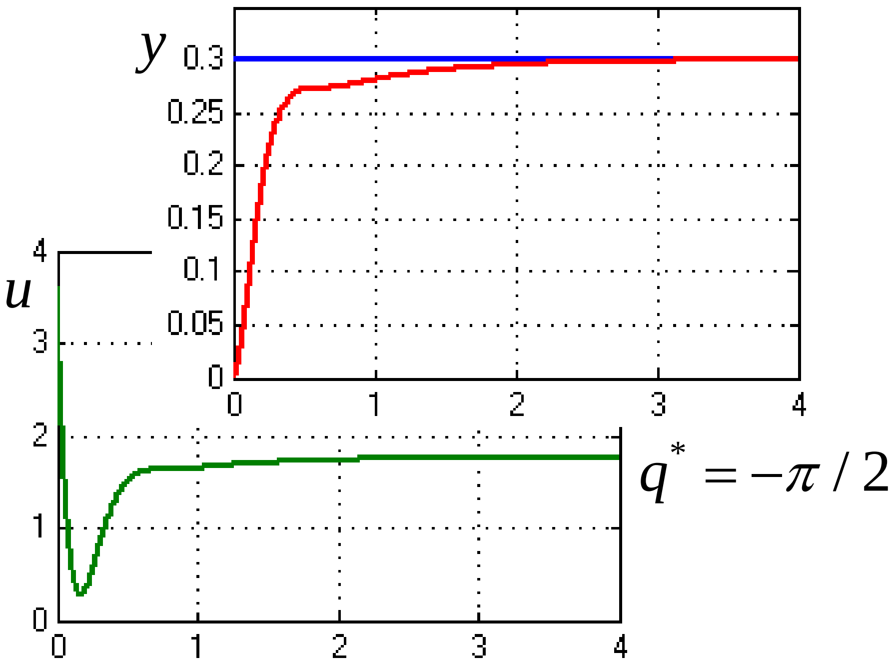
- Slowest poles take about 4 s to converge
- Because of the **repeated poles**, they become **complex conjugate** (pair) when one parameter is varied

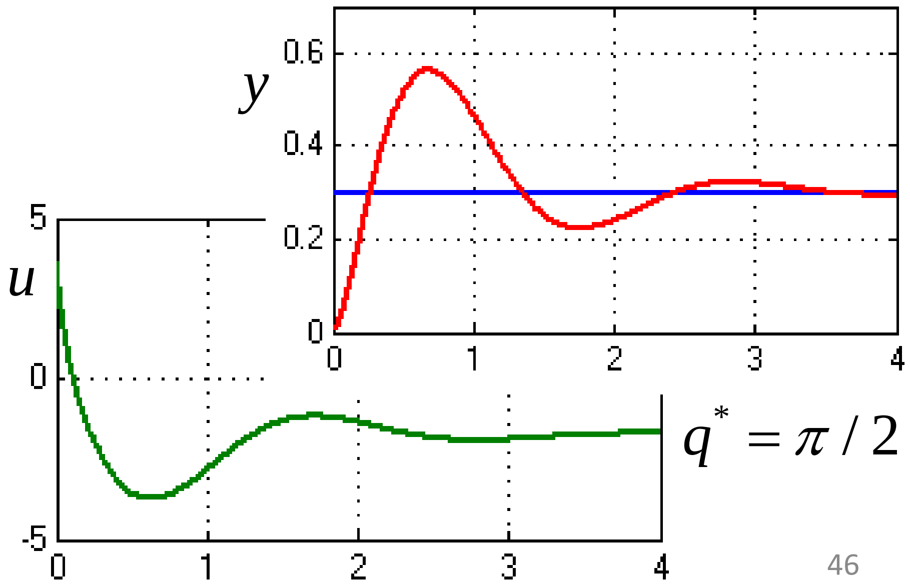
PID Simulations 1

A 0.3 rad step demand for the 3 linear systems described earlier

- Considerable variation in transient response shapes
- Considerable overshoot (zero) for equilibrium point $q^* = 0$
- Slower settling for non equilibrium points $q^* = \pm\pi/2$







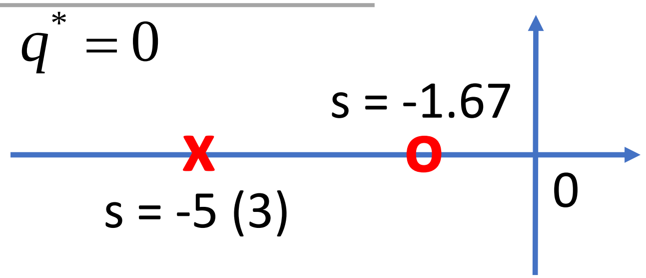
Pole – Zero Plot Analysis 1

Pole-zero plots can give us an idea about the effect of the zero(s) on the dynamic response.

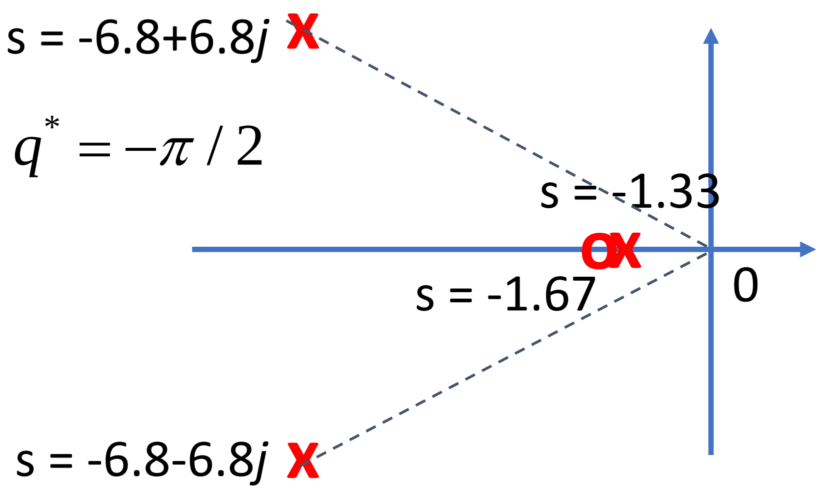
Consider the pole – zero plots for the previous 3 simulations.

Note the zero remains unchanged

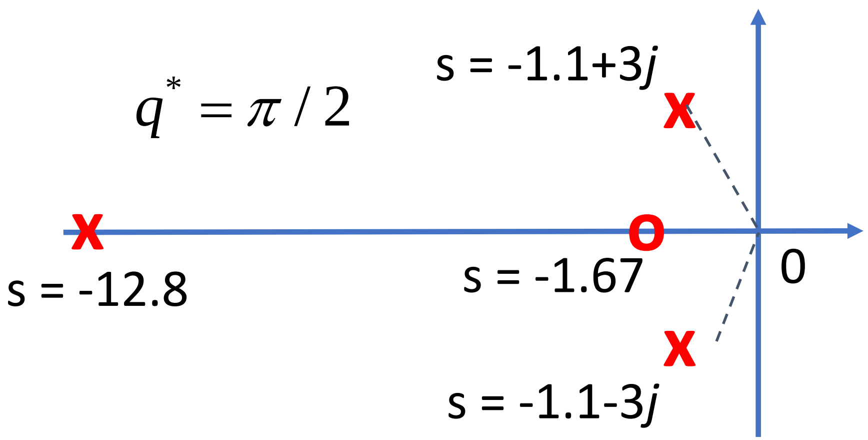
Large variation in plots



Real, negative zero, distinct from the 3 repeated poles, which means that there will be overshoot



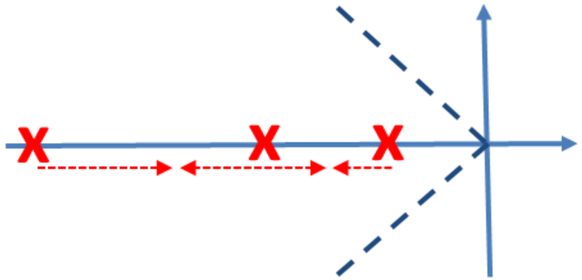
Two (conjugate) mildly oscillatory poles (w~s) are fast. Slow, real pole is close to the zero so contribution is reduced and little overshoot



Two (conjugate) complex poles where w>s and distinct from the real zero, so an oscillatory response (largely 2nd order) is expected

Real, Distinct Pole Placement 2

- Consider specifying distinct, real poles so that the parameter variation does not cause a significant oscillations in the response



- To avoid oscillations, but to not demand too much control energy, let us try poles at s = -5, -10 and -20. This leaves an interval on the real axis (s-plane) for the poles to move, due to parameter (linearization point) variation

- Assume $q^* = 0$
- Then

$$(s+5)(s+10)(s+20)=0$$

$$s^3+35s^2+350s+1000=0$$

Therefore

$$s^3+6.25k_D s^2+([-36.75,36.75]+6.25k_P)s+6.25k_I=0$$

$$k_P=56, \quad k_I=160, \quad k_D=5.6$$

Faster response required => increased controller gains / control energy

Specifying Time Constants / Poles

- Vertical upright: $q^* = \pi/2$,

$$s^3 + 35s^2 + 313.25s + 1000 = 0$$

$$s = \{-23.77, -5.77 \pm 3.06j\}$$

- Vertical down: $q^* = -\pi/2$,

$$s^3 + 35s^2 + 386.75s + 1000 = 0$$

$$s = \{-3.69, -15.66 \pm 5.12j\}$$

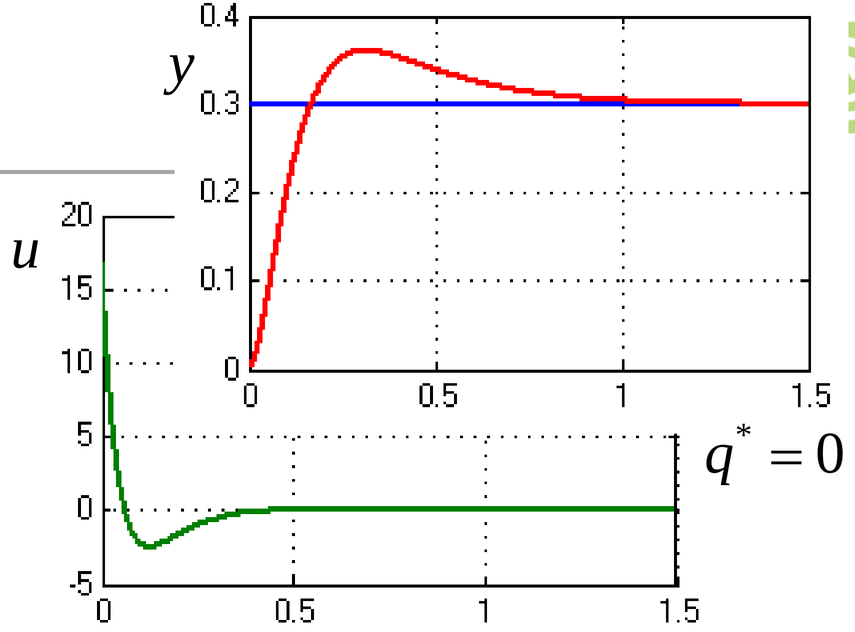
- Slowest poles take about 1 s to converge
- Oscillatory poles are closer to the real axis, so oscillations are less obvious
- Control gains are larger

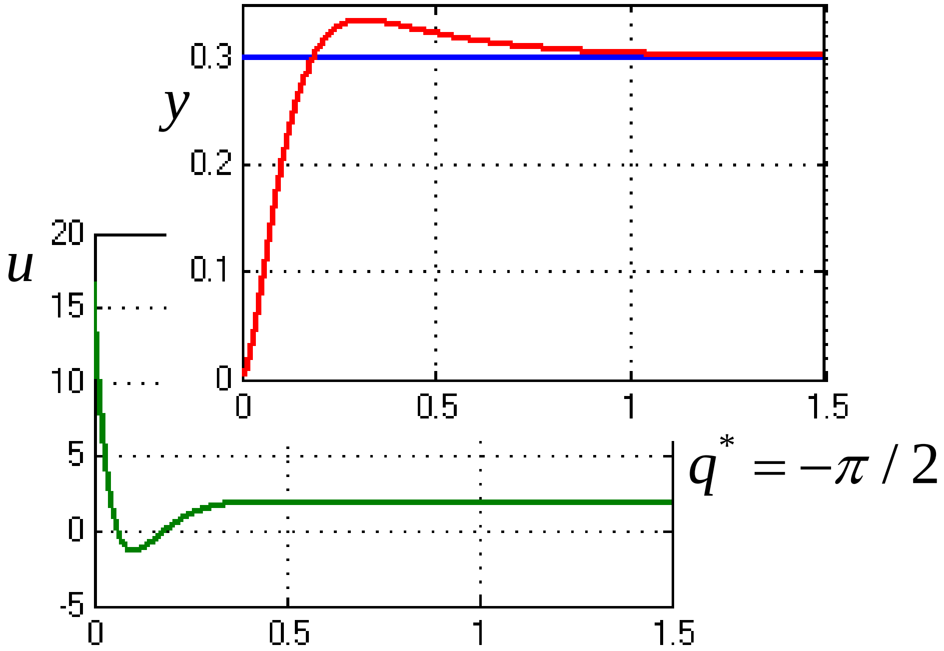
- A key observation about why this may be a "better" solution is that the coefficient is much larger (approximately 5 times, 350 compared to 75), hence the effect of the variation by ±35 is less.
- While it may be better in terms of reducing the effect of parameter variation, this is done by increasing the controller gains and hence the energy used by the controller.
- Physically, using more control energy reduces the relative effect / variation by gravity

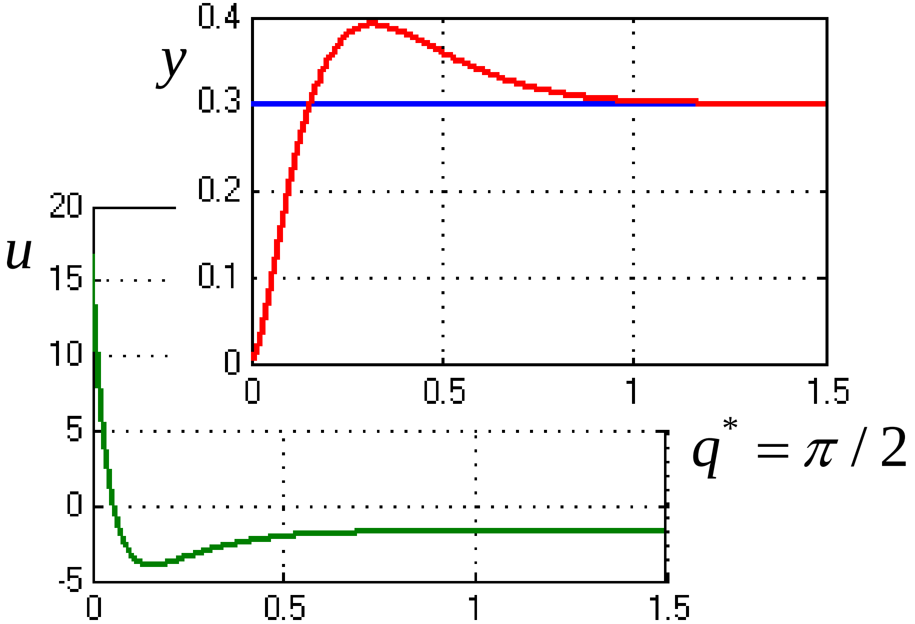
PID Simulations 2

Same simulation conditions, but with modified PID gains

- Settling time is ~1 s in all 3 cases and similar shapes
- Oscillatory modes are difficult to see
- Peak control magnitude is about 4 times greater







Pole - Zero Analysis 2

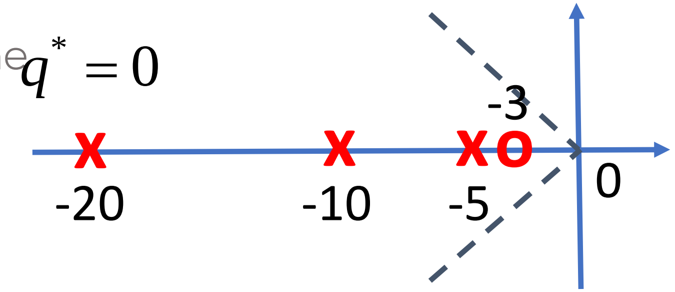
Pole-zero plots can give us an idea about the effect of the zero(s) on the dynamic response.

Consider the pole – zero plots for the previous 3 simulations.

Note the zero remains unchanged

Large variation in plots

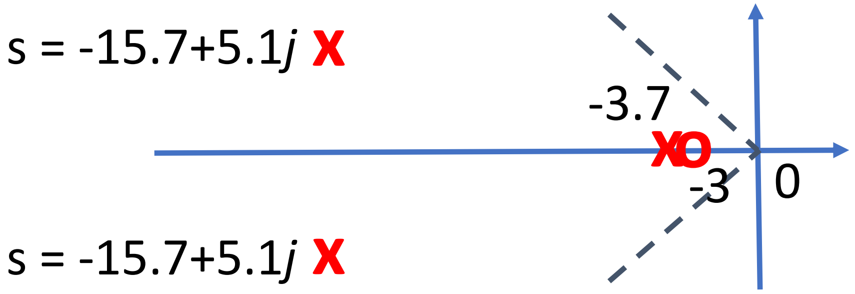
$$q^* = 0$$



Real, negative zero, distinct from the 3 repeated poles, which means that there will be overshoot.

Zero "close" to slowest pole

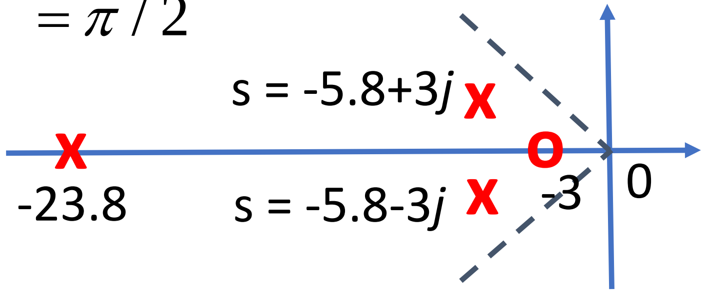
$$q^* = -\pi / 2$$



Two fast, conjugate poles with small oscillations (w<s) are fast.

Slow, real pole is close to the zero so contribution is reduced and little overshoot

$$q^* = \pi / 2$$



Two slower, conjugate poles with small oscillations (w<s).

Poles are distinct from the real zero, so some overshoot is expected. Fast (negligible) real pole.

SLM Model-based Control

We'll explicitly use knowledge about the non-linear dynamics to design global, high performing controllers

$$(J + ma^2)\ddot{q} + mga\cos(q) = \tau$$

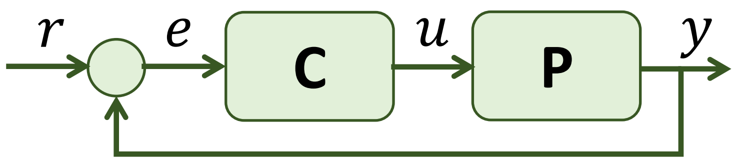
- **Model-based** refers to the fact that the controller has a perfect (or near perfect) model which requires both accurate knowledge of physical parameters (masses, lengths, …) & physical processes (friction, stiction, gear-box, …)
- Improve **performance** which is important for developing new applications & capabilities as well as differentiate offerings

This section will cover:

1. Gravity compensation
2. Computer torque control

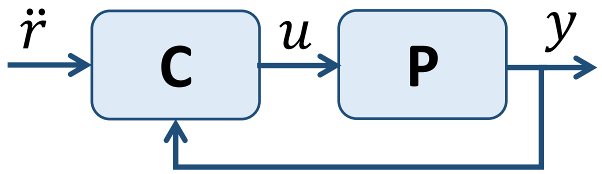
Model-based verses Feedback Control

Both model-based and feedback control techniques attempt to modify the plant dynamics so that the closed loop system tracks a reference signal, although they're very different approaches



Feedback control

- Designed to be insensitive to variations in the plant parameters
- Control signal is reactive, it takes time to "integrate up" the error to compute the appropriate steady state signal

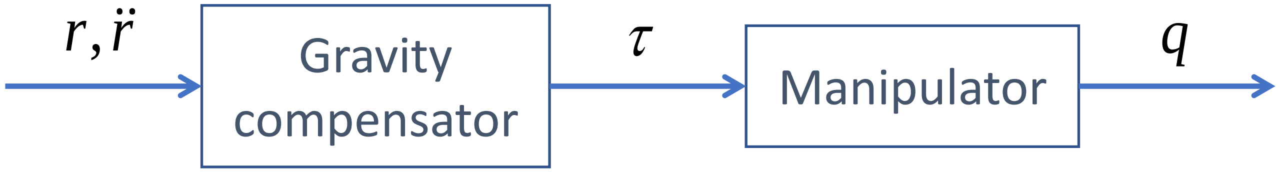


Model-based control

- Assumes full knowledge of the (non-linear) dynamics
- High performance, action is a function of current state & desired reference

In practice, a real-life controller may make use of both approaches

Gravity Compensation



Assuming we know the desired joint acceleration (kinematics) and have perfect plant knowledge (parameters), the required torque is:

$$\tau = \boxed{(J + ma^2)\ddot{r}} + \boxed{mga\cos(r)}$$

reference acceleration *counteract gravity*

Combining the plant dynamics and the gravity compensator:

$$(J + ma^2)\ddot{q} + mga\cos(q) = (J + ma^2)\ddot{r} + mga\cos(r)$$

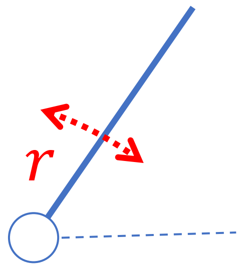
$$\ddot{q} = \ddot{r}$$

Assuming no initial errors, i.e. $q(0) = r(0), \dot{q}(0) = \dot{r}(0)$, the manipulator's position $q(t)$ follows the desired reference perfectly.

Example: Gravity Compensation

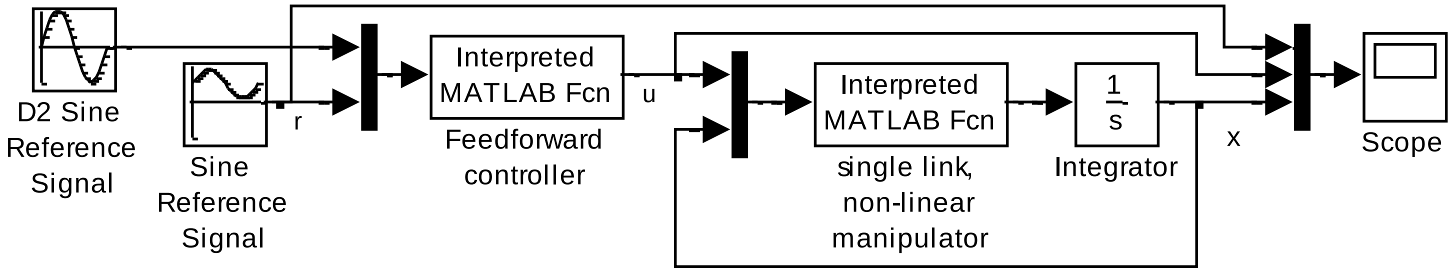
Consider the non-linear, sinusoidal tracking problem just described, but using only gravity compensation "control"

- Assume perfect knowledge of the plant parameters
- Assume zero initial errors (considered later)
- Assume perfect differentiation (acceleration)



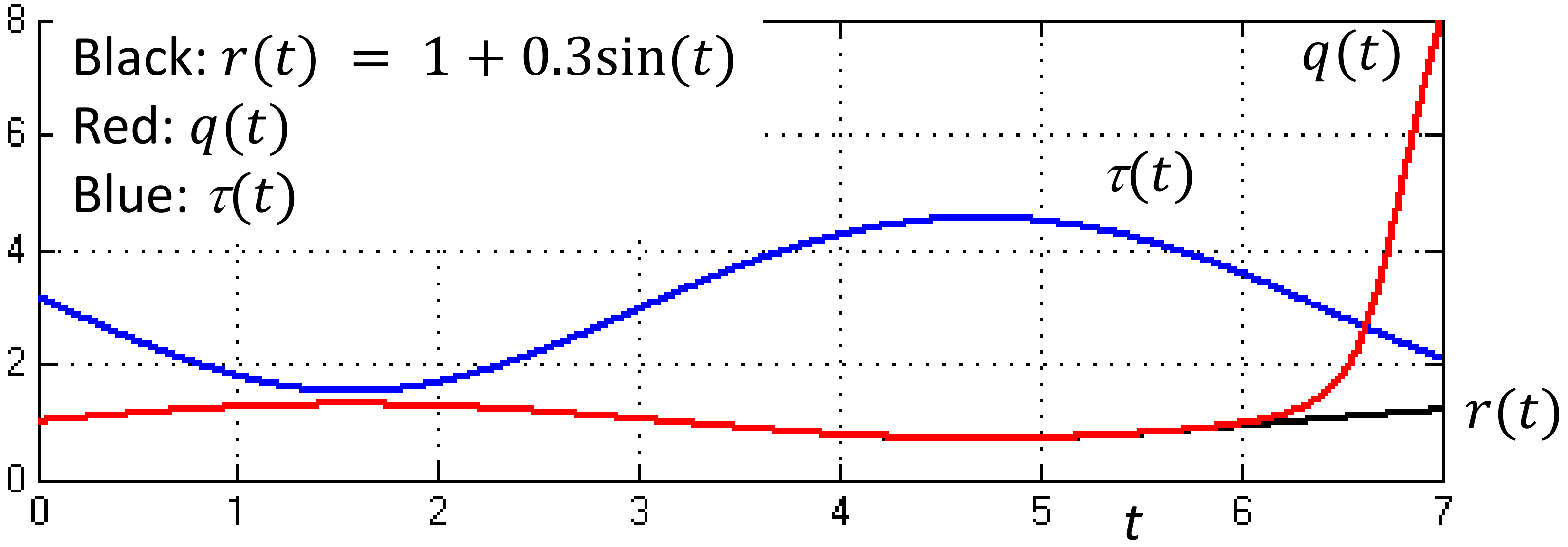
(none of this is realistic in real-life)

Simulink example (normal, exemplar manipulator)



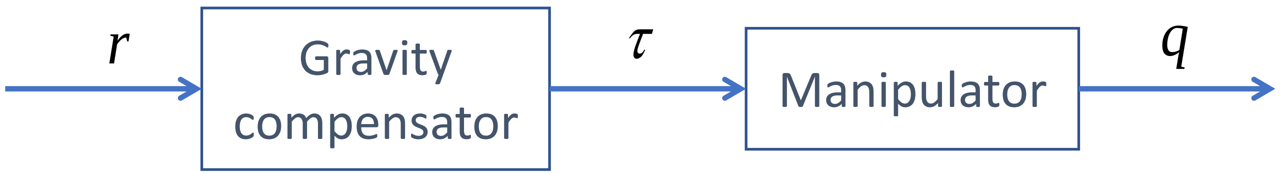
- Simulink blocks for the manipulator $\ddot{q} = 6.25\tau - 36.75\cos(q)$ and the controller $\tau = 0.16\ddot{r} + 5.88\cos(r)$
- $r = 1 + 0.3\sin(t)$, so the torque signal should vary
- $q(0) = 1, \dot{q}(0) = 0.3$ which means zero initial errors

Gravity Compensation Results



- Until about 6 s, the joint angle, q, appears to perfectly track the sinusoidal reference, r, using the feedforward controller
- The calculated torque is approximately sinusoidal as well, but out of phase by π because it is a function of the reference acceleration
- After 6 s, there is a visual exponential divergence between the joint and reference signals
- Caused by tiny "errors" in the numerical integration (or other sources) which exponentially grow in magnitude

5.1 Gravity Compensation Analysis



$$(J + ma^2)\ddot{q} + mga\cos(q) = mga\cos(r) + (J + ma^2)\ddot{r}$$

linearized

$$\Delta\ddot{q} - \frac{mga\sin(q^*)}{(J + ma^2)}\Delta q = \frac{mga\cos(r^*)}{(J + ma^2)}\Delta r + \Delta\ddot{r}$$

Gravity compensation is a function of the reference, r, and is independent of the manipulator's state (feedforward). So the poles of the overall $r \rightarrow q$ mapping are just the poles of the manipulator.

From the previous analysis:

- Above the horizontal the manipulator is unstable (one positive pole) and any error will locally exponentially diverge (similar to torque balance, section 3.1)
- Below the horizontal the manipulator is marginally stable (imaginary poles) and any error will persist

Computer Torque Control

The previous model-based control strategy means that the torque is not robust to errors (initial state errors, disturbances, …)

Consider two modifications:

- Include an **additional** (feedback error) **signal** v and
- The **gravity component** uses the **current state** q:

This is called **Computer Torque Control (CTC)** and the dynamics are:

$$\tau = (J + ma^2)(\ddot{r} + v) + mga\cos(q)$$

The resulting "closed loop", $r \rightarrow q$ dynamics are:

$$(J + ma^2)\ddot{q} + mga\cos(q) = (J + ma^2)(\ddot{r} + v) + mga\cos(q)$$

$$\ddot{q} = \ddot{r} + v \qquad \ddot{e} + v = 0$$

The joint error, $e = r - q$, dynamics doesn't depend on the manipulator's dynamics, rather they are determined by the design of v.

This is a simple example of **feedback linearization**

CTC Error Dynamics

For the error dynamics (ODE) $\ddot{e} + v = 0$, consider v defined by:

$$v = k_D \dot{e} + k_P e$$

for user defined values of k_P and k_D. Then the ODE becomes

$$\ddot{e} + k_D \dot{e} + k_P e = 0$$

- The error dynamics are linear, unforced, 2nd order
- The error dynamics (poles) are determined by choosing k_P and k_D
- The error dynamics are independent of the non-linear dynamics, a major difference to conventional feedback control

In robotics, a **critically damped response** is typically used:

- Two equal poles with a time constant (inverse pole(s)) about ¼ of desired settling time.
- Critical damping means **no overshoot** in the joint angles
- **Minimize peak torque** because there is no “fast” pole

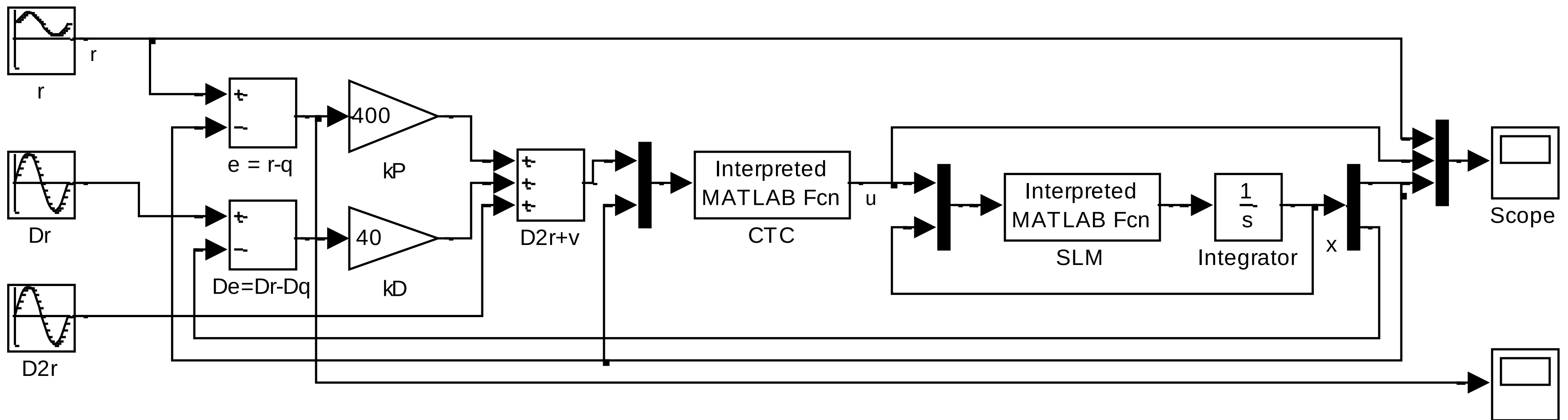
Example: CTC Joint Control

For a settling time of about 0.2 s, use a time constant of τ=0.05 s (poles=-20) and the 2nd order error ODE is:

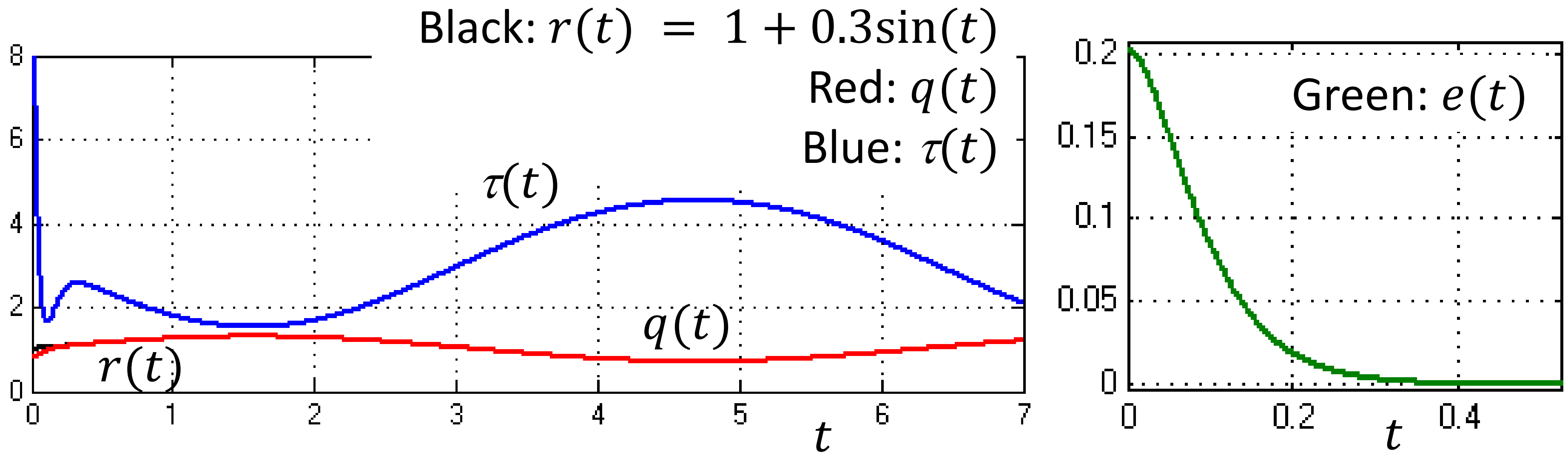
$$(s+20)(s+20)=0$$
$$s^2 + \boxed{40s} + \boxed{400} = 0$$

$$k_D = 40$$
$$k_P = 400$$

For the usual, exemplar manipulator $\ddot{q} + 36.75\cos(q) = 6.25\tau$ and CTC $\tau = 0.16(\ddot{r} + v) + 5.88\cos(q)$ with the sinusoidal reference signal $r = 1 + 0.3\sin(t)$ and an initial state $q(0) = 0.8,\ \dot{q}(0) = 0$. This produces initial joint errors $e(0) = 0.2,\ \dot{e}(0) = 0.3$.



Example: CTC Joint Control Results



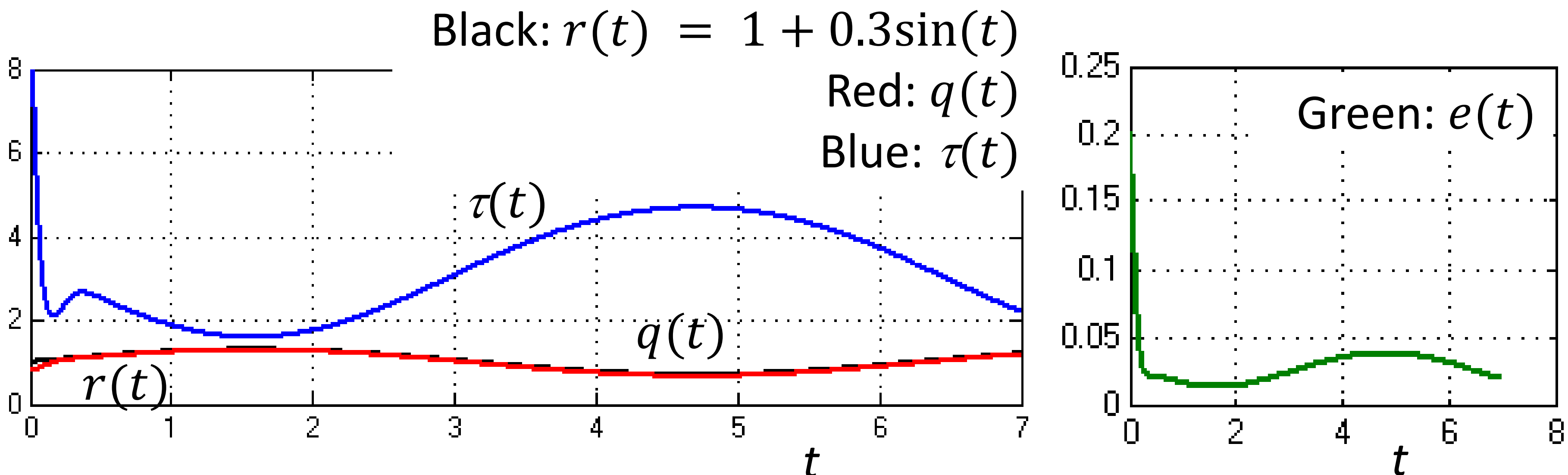
- The CTC successfully generates a torque which both eliminates the initial joint error after about 0.3 s and thereafter accurately tracks the joint reference signal
- The errors converge to zero as a critically damped, 2nd order ODE
- With zero error, the CTC simply solves the equation of motion for the reference signal
- We still assume the equation of motion is known exactly

CTC Effects of Model Errors

Consider when the previous simulation is repeated, except that the CTC controller is based on **inaccurate** parameter values

$$\tau = 0.16(\ddot{r} + v) + 5.88\cos(q) \quad \Rightarrow \quad \tau = 0.1(\ddot{r} + v) + 4\cos(q)$$

- Tracking performance is still reasonable
- Errors **don't** converge to zero as there is a non-zero excitation term in the error dynamics
- Possible to design additional robust elements



Model-based Control Summary

- Model-based control assumes there is accurate knowledge of the model's dynamics and a (twice) continuously differentiable reference signal and uses this directly to compute the torque
- A key aim (with manipulators) is to compensate for gravity in a feedforward, instantaneous manner rather than having to integrate up the error as occurs with feedback control
- An additional element determines the torque required to produce a reference acceleration
- Feedforward gravity compensation does not alter the dynamics (poles) of the manipulator and hence is still unstable
- CTC introduces an additional error feedback signal which decouples the error's dynamics from the manipulator's dynamics and this can be designed to be stable
- The typical CTC response is critical damped as this produces no overshoot (no zeros) and minimal peak torque