Dual Link Manipulator

Dynamical Model

{Learn, Create, Innovate};

Dual Link Manipulator

Introduction

- For this section, we'll concentrate on a simple *2-link manipulator.*
- Some of the equations & analyses are surprisingly complex.
- Is one of the most used configurations in the industry.
- Can also be used to model legged robots.
- Exhibits many of the basic locomotion problems associated with more complex skeleton models

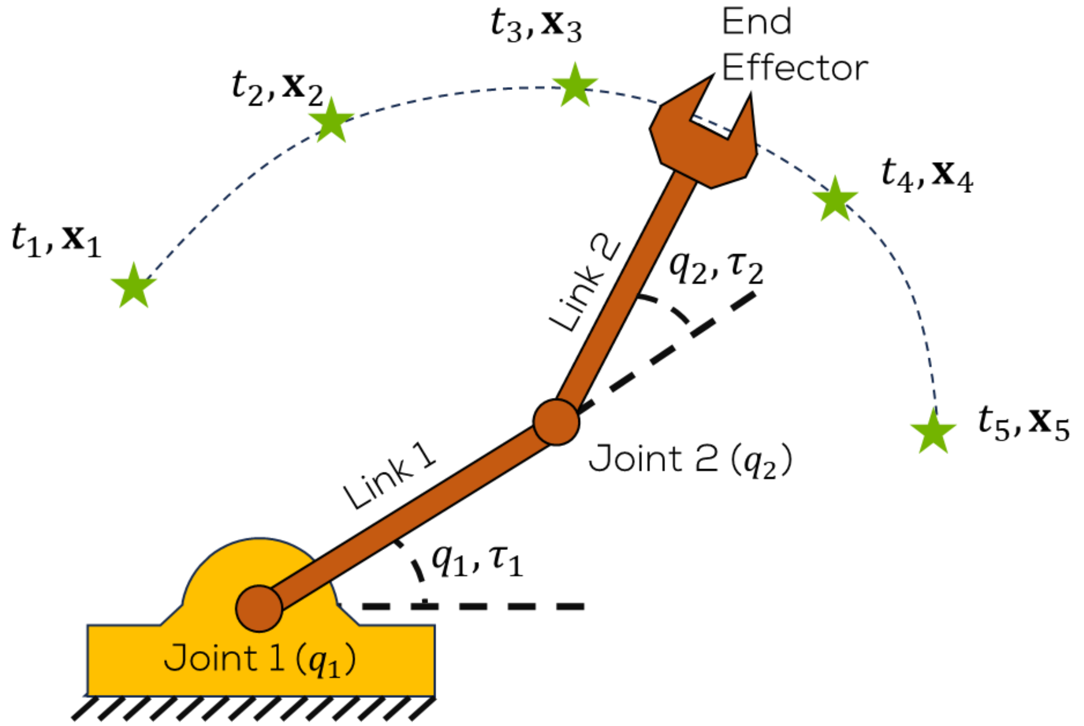
IRB_5720 (ABB)

ASIMO (Honda, Japan)

Dual Link Manipulator

: eQ\BY^[C Q^Ÿ e\Qd_b¥ BC¦

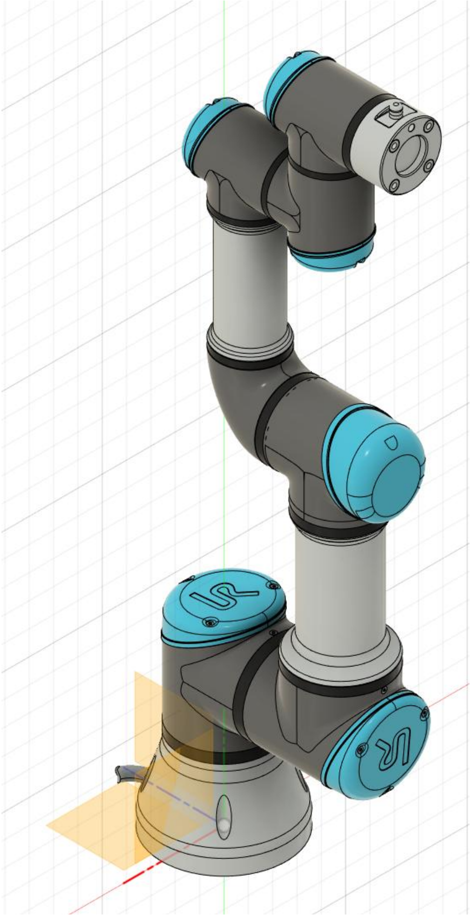
- Composed of 2 links and 2 joints
- Joints are referenced to the horizontal.
- We want to follow a trajectory with the end effector.
- As with the SLM, a motor in each joint is usually used to move the links to a desired position.



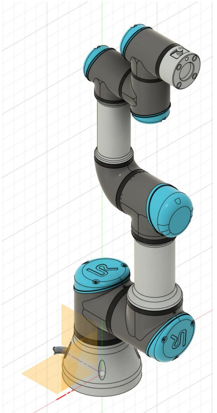
Dual Link Manipulator

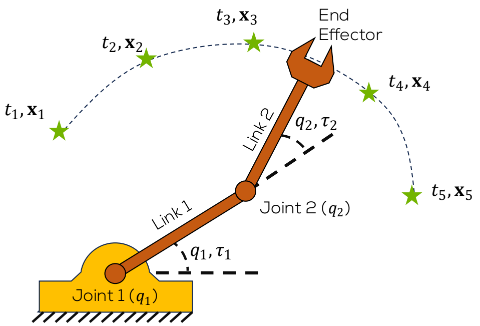
Dual Link Manipulator

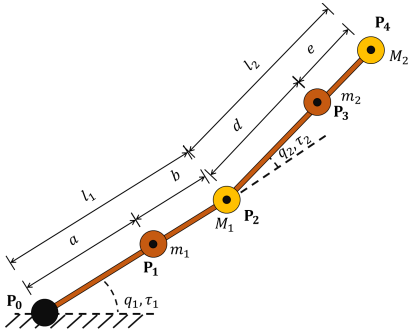
- As for the previous SLM, the dynamics will be obtained using the Euler-Lagrange approach.
- This analysis aims to enhance the comprehension of the intricate behaviour of the system and may be of interest to professionals working in this field.
- Several levels of abstraction will be performed.



DLM Modelling





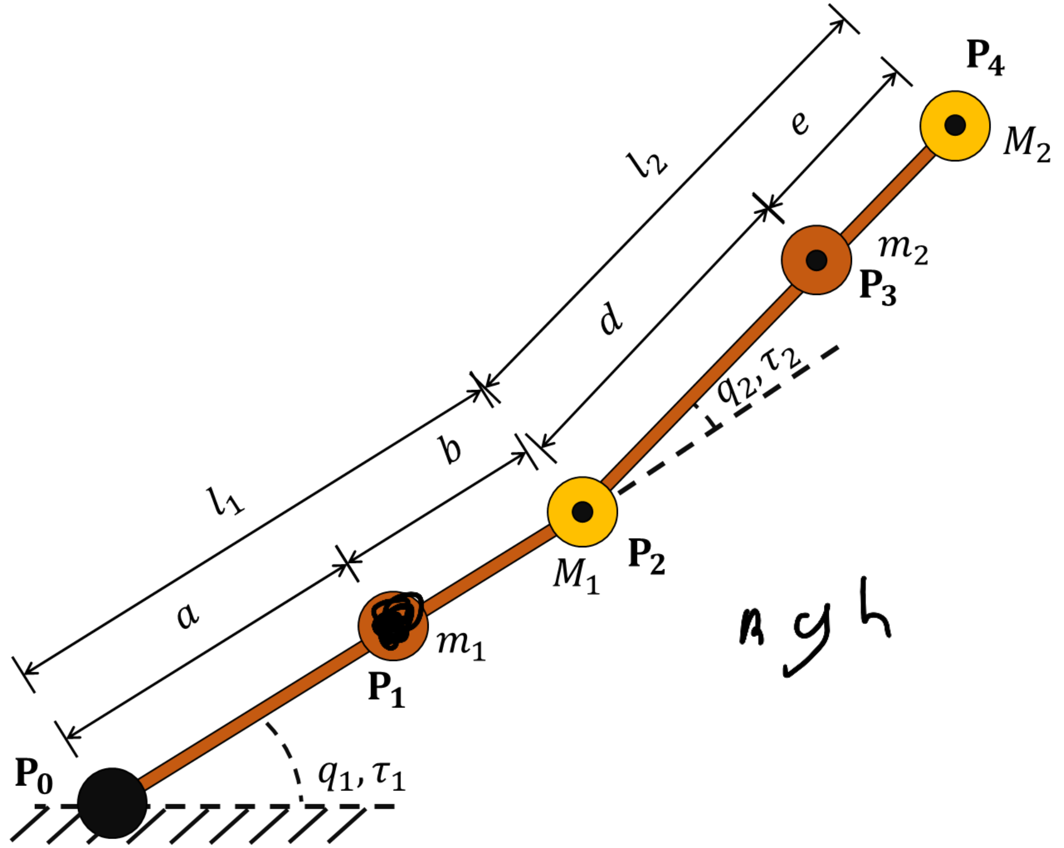


$$\boldsymbol{\tau} \rightarrow \boxed{\ddot{q} = f(q, \dot{q}, \tau)} \rightarrow \boldsymbol{q}$$

Dual Link Manipulator

C _TU\

- l_1, l_2 Links length
  - $l_1 = a + b, \quad l_2 = d + e$
- m_1, m_2 = link's mass
- M_1 =Motor mass
- M_2 =Load mass
- q_1 =angle relative to the floor
- q_2 =relative joint angle
- $\boldsymbol{P}_i$ =Cartesian position of point (COM)



DLM Modelling

- Euler Lagrange Equation

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

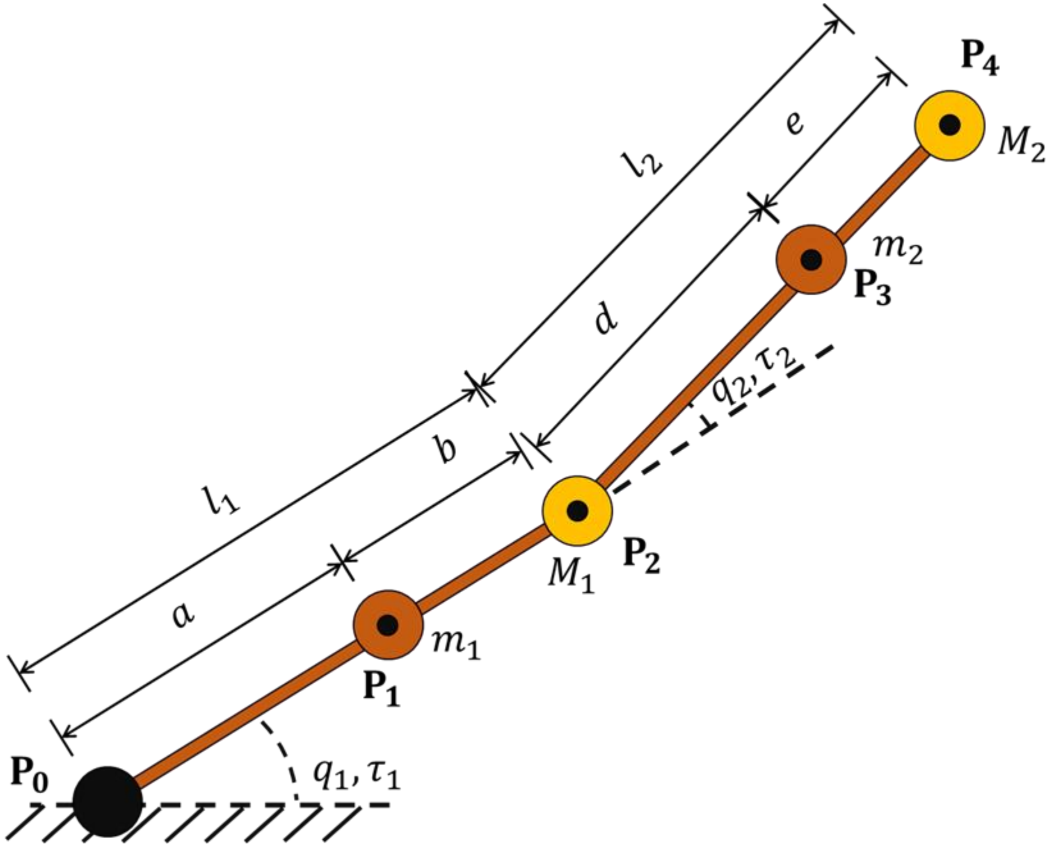
where

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

then

- $\frac{\partial L(q,\dot{q})}{\partial q}=\frac{\partial}{\partial q}\big(KE(q,\dot{q})-PE(q)\big)\Rightarrow\frac{\partial}{\partial q}KE(q,\dot{q}))-\frac{\partial}{\partial q}PE(q)$
- $\frac{\partial L(q,\dot{q})}{\partial \dot{q}}=\frac{\partial}{\partial \dot{q}}KE(q,\dot{q})$

$$\therefore \frac{d}{dt}\left(\frac{\partial}{\partial \dot{q}}KE(q,\dot{q})\right)-\frac{\partial}{\partial q}KE(q,\dot{q}))+\frac{\partial}{\partial q}PE(q)=\tau$$



DLM Modelling

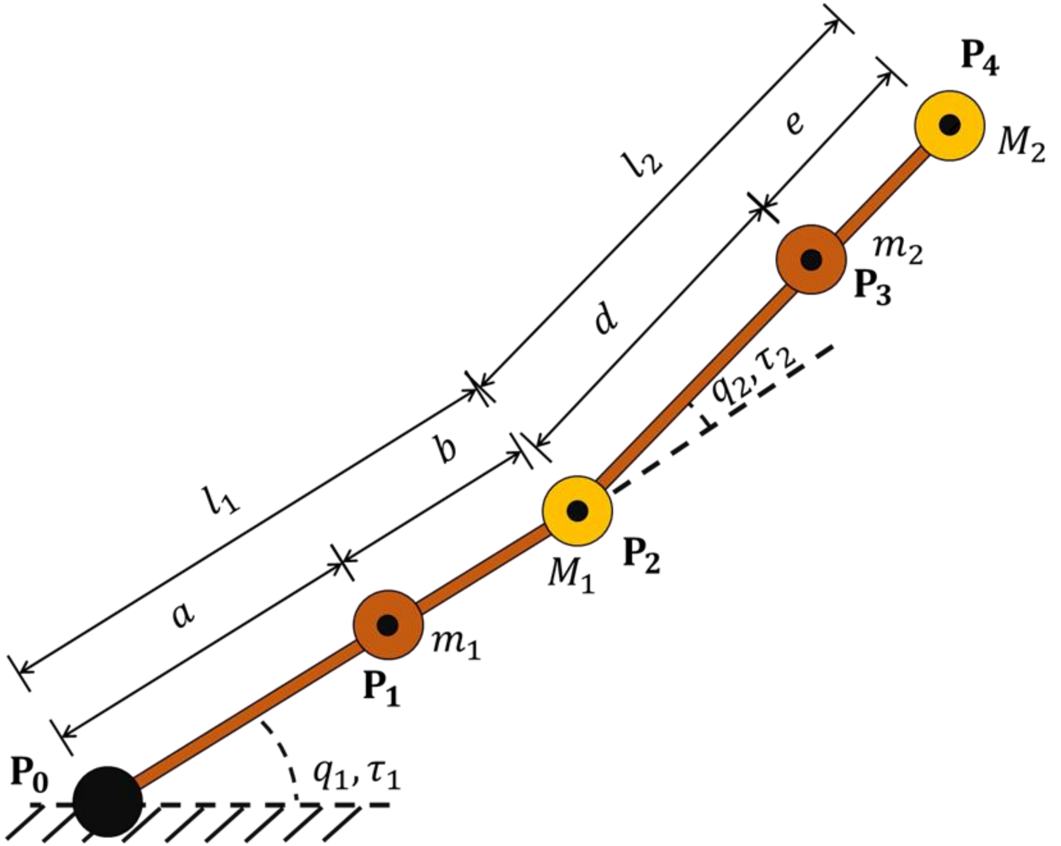
DLM Manipulator (Forward Kinematics)

- The KE and PE depend on the Cartesian **position** and **velocities** of the point masses.
- The velocities of the point masses are required to be expressed in joint space to evaluate the Euler-Lagrange expression.
- The forward's kinematics are:

$$\boldsymbol{P}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{P}_1 = a \begin{bmatrix} \cos(q_1) \\ \sin(q_1) \end{bmatrix}, \quad \boldsymbol{P}_2 = l_1 \begin{bmatrix} \cos(q_1) \\ \sin(q_1) \end{bmatrix}$$

$$\boldsymbol{P}_3 = l_1 \begin{bmatrix} \cos(q_1) \\ \sin(q_1) \end{bmatrix} + d \begin{bmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{bmatrix}$$

$$\boldsymbol{P}_4 = l_1 \begin{bmatrix} \cos(q_1) \\ \sin(q_1) \end{bmatrix} + l_2 \begin{bmatrix} \cos(q_1 + q_2) \\ \sin(q_1 + q_2) \end{bmatrix}$$



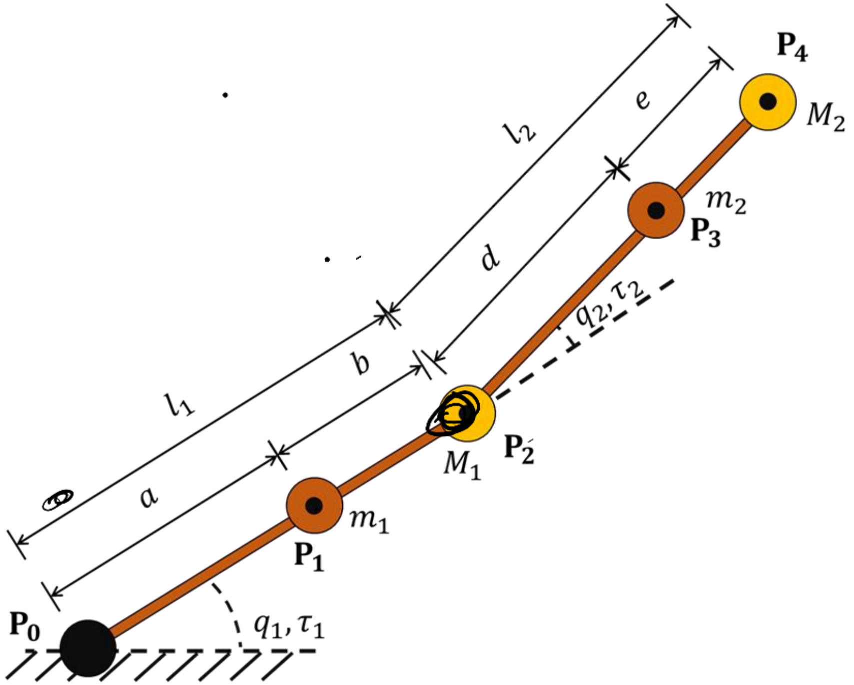
DLM Modelling

: BC C Q^Y e\Qd_b¥F_dU^dYQ\U^UbWi¦

- The $PE(\boldsymbol{q})$ depends on the cartesian position of the mass distribution

$$PE(\boldsymbol{q}) = g\Sigma m_i h_i \Rightarrow g\sum m_i P_{i,y}$$

$$PE(q) = g\big(m_1 a S_1 + M_1 l_1 S_1 + m_2(l_1 S_1 + d S_{12}) + M_2(l_1 S_1 + l_2 S_{12})\big)$$



- According to Euler-Lagrange we need:

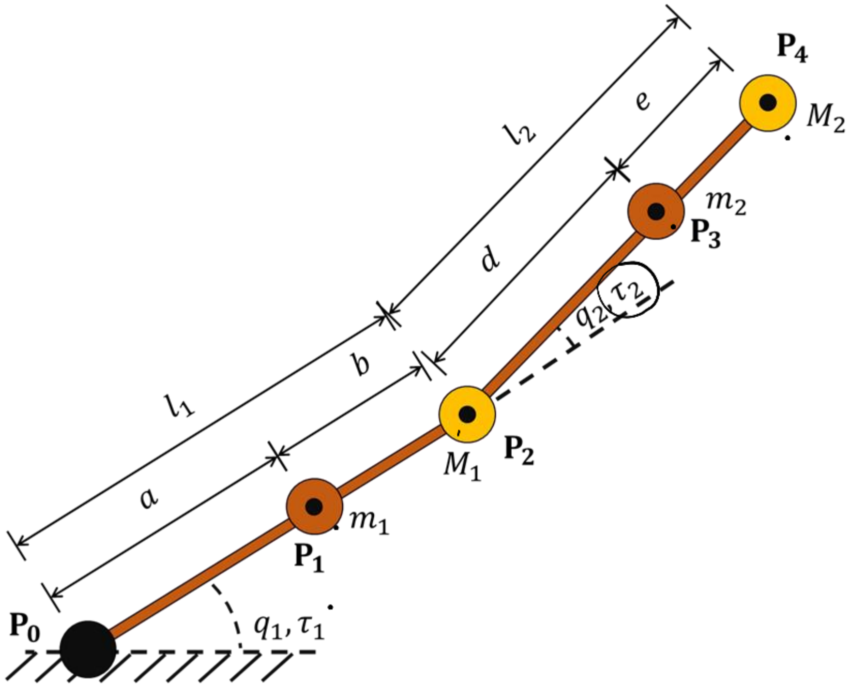
$$\frac{\partial PE(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{bmatrix} \dfrac{\partial PE(\boldsymbol{q})}{\partial q_1} \\ \dfrac{\partial PE(\boldsymbol{q})}{\partial q_2} \end{bmatrix}$$

$$\frac{\partial PE(\boldsymbol{q})}{\partial \boldsymbol{q}} = g\begin{bmatrix} m_1 a C_1 + M_1 l_1 C_1 + m_2(l_1 C_1 + d C_{12}) + M_2(l_1 C_1 + l_2 C_{12}) \\ C_{12}(m_2 d + M_2 l_2) \end{bmatrix}$$

DLM Modelling

: BC C Q^Y e\Qd_b¥F_dU^dYQ\U^UbWi¦

$$\frac{\partial PE(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{g}(\boldsymbol{q}) = g\begin{bmatrix} m_1aC_1 + M_1l_1C_1 + m_2(l_1C_1 + dC_{12}) + M_2(l_1C_1 + l_2C_{12}) \\ C_{12}(m_2d + M_2l_2) \end{bmatrix}$$



- It is termed the **gravity vector** in the equations of motion & represents the torque on each joint due to gravity acting on the point masses.
- It is the **equilibrium torques** (acting at the joints) necessary to hold the "manipulator" at zero acceleration (velocity)

DLM Modelling

DLM Manipulator (Velocity Kinematics)

- The KE depends on the Cartesian **velocities** of the point masses which we need to express in joint space $\{\mathbf{q}, \dot{\mathbf{q}}\}$.
- Let $\sin(q_i) = s_i,\ \cos(q_i) = c_i, \sin(a+b) = s_{ab}$ and $\cos(a+b) = c_{ab}$

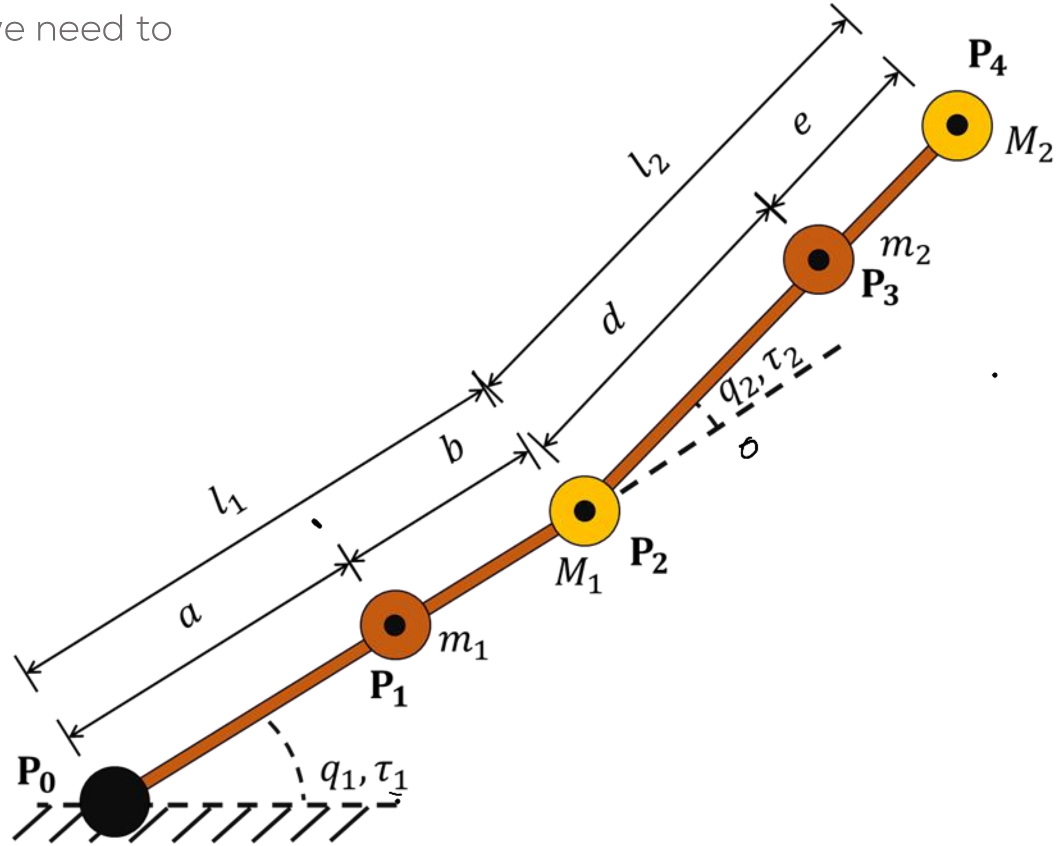
$$\dot{P}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$\dot{P}_1 = a\begin{bmatrix} -\sin(q_1) \\ \cos(q_1) \end{bmatrix}\dot{q} = a\begin{bmatrix} -s_1 \\ c_1 \end{bmatrix}\dot{q} = a\begin{bmatrix} -s_1 & 0 \\ c_1 & 0 \end{bmatrix}\dot{\boldsymbol{q}}$$

$$\dot{P}_2 = l_1\begin{bmatrix} \cos(q_1) \\ \sin(q_1) \end{bmatrix} = l_1\begin{bmatrix} -s_1 \\ c_1 \end{bmatrix}\dot{q} = l_1\begin{bmatrix} -s_1 & 0 \\ c_1 & 0 \end{bmatrix}\dot{\boldsymbol{q}}$$

$$\dot{P}_3 = \begin{bmatrix} -(l_1 s_1 + d s_{12}) & -d s_{12} \\ l_1 c_1 + d c_{12} & d c_{12} \end{bmatrix}\dot{\boldsymbol{q}}$$

$$\dot{P}_4 = \begin{bmatrix} -(l_1 s_1 + l_2 s_{12}) & -l_2 s_{12} \\ l_1 c_1 + l_2 c_{12} & l_2 c_{12} \end{bmatrix}\dot{\boldsymbol{q}}$$



DLM Manipulator (Kinetic Energy)

- The KE can then be described as

$$KE(\boldsymbol{q},\dot{\boldsymbol{q}}) = \frac{1}{2}\sum m_i v_i^2 + \frac{1}{2}\sum J_i \dot{q}_i$$

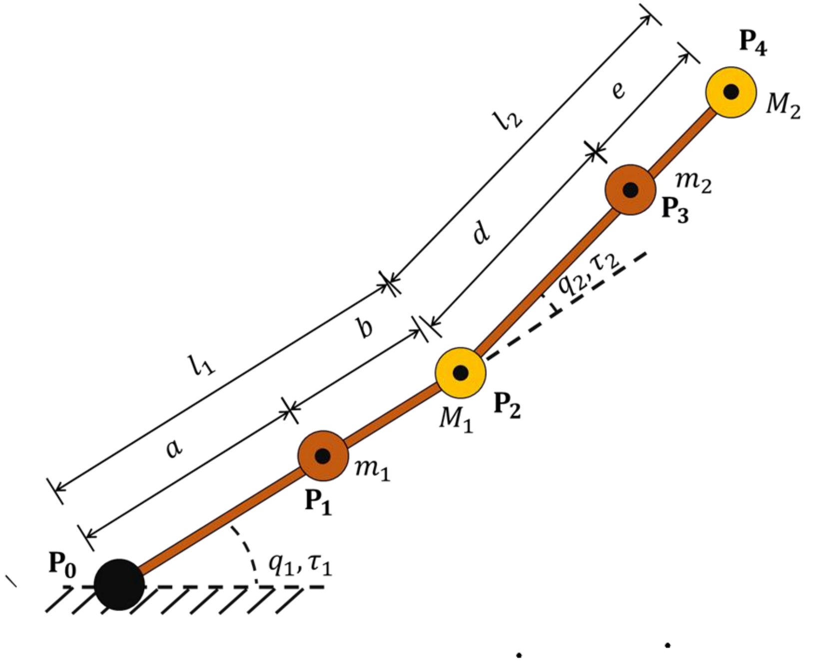
- For this case, we assume the inertias of the links Is neglected to make a simpler model.

$$KE(\boldsymbol{q},\dot{\boldsymbol{q}}) = \frac{1}{2}\left(m_1\|\dot{\boldsymbol{P}}_1\|_2^2 + M_1\|\dot{\boldsymbol{P}}_2\|_2^2 + m_2\|\dot{\boldsymbol{P}}_3\|_2^2 + M_2\|\dot{\boldsymbol{P}}_4\|_2^2\right)$$

$$KE(\boldsymbol{q},\dot{\boldsymbol{q}})$$

$$= \frac{1}{2}\bigg(\Big(m_1a^2 + M_1l_1^2 + m_2(l_1^2 + d^2 + 2l_1dC_2) + M_2(l_1^2 + l_2^2 + 2l_1l_2C_2)\Big)\dot{q}_1^2$$

$$+ 2\big(m_2d(l_1C_2 + d) + M_2l_2(l_1C_2 + l_2)\big)\dot{q}_1\dot{q}_2 + \big(m_2d^2 + M_2l_2^2\big)\dot{q}_2^2\bigg)$$



DLM Modelling

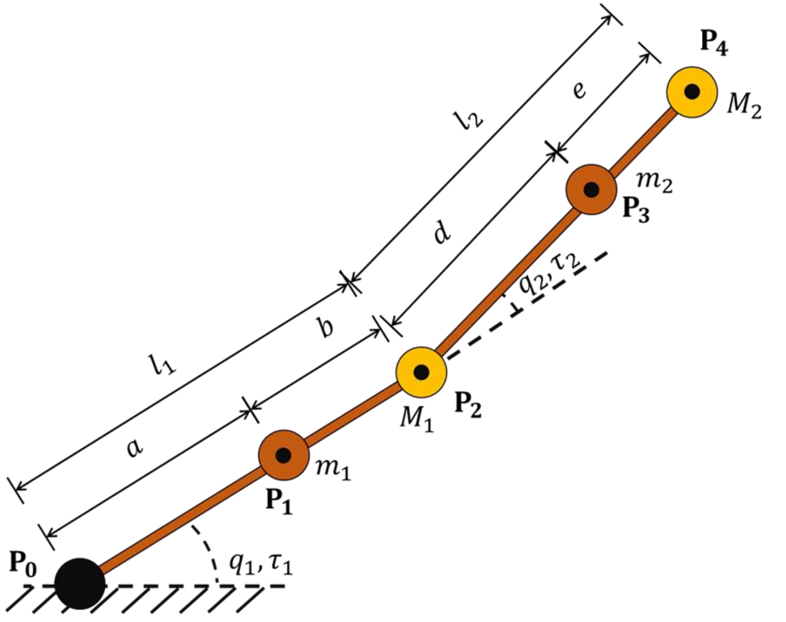
: BC C Q^Y e\Qd_b¥AY^UdS; ^UbWi¦

- Expressing the KE in terms of the angular velocities:
- This is **quadratic** in the **angular velocities** (*states*)
- The coefficients are functions of mass, length and the inter-leg angle q_2, it is independent of q_1.
- The KE can then be rewritten as

$$KE(\boldsymbol{q},\dot{\boldsymbol{q}}) = \frac{1}{2}\,\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Where

$$M(\boldsymbol{q}) = \begin{bmatrix} m_1a^2 + M_1l_1^2 + m_2(l_1^2 + d^2 + 2l_1dC_2) + M_2(l_1^2 + l_2^2 + 2l_1l_2C_2) & m_2d(l_1C_2 + d) + M_2l_2(l_1C_2 + l_2) \\ m_2d(l_1C_2 + d) + M_2l_2(l_1C_2 + l_2) & m_2d^2 + M_2l_2^2 \end{bmatrix}$$

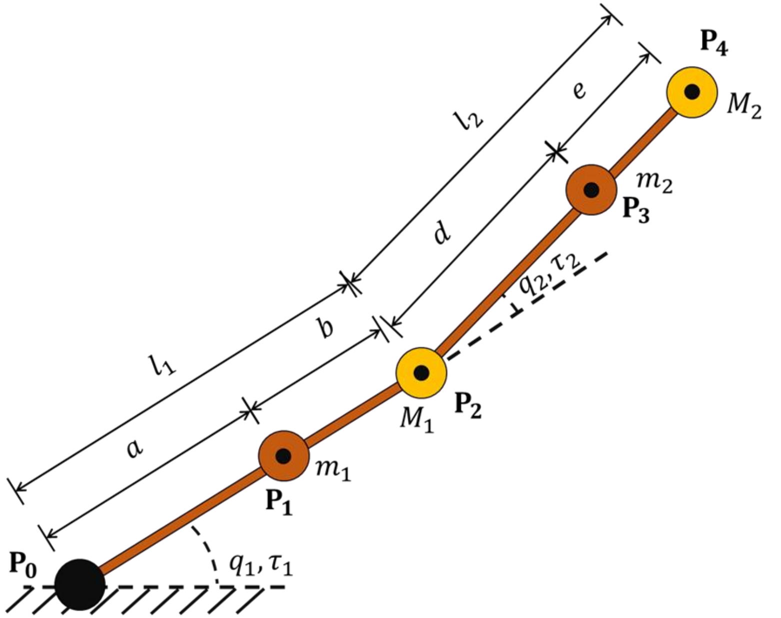


DLM Modelling

: BC C Q^Ÿ e\Qd_b¥AY^UdS; ^UbWï¦

$$M(\boldsymbol{q}) = \begin{bmatrix} m_1 a^2 + M_1 l_1^2 + m_2(l_1^2 + d^2 + 2 l_1 d C_2) + M_2(l_1^2 + l_2^2 + 2 l_1 l_2 C_2) & m_2 d(l_1 C_2 + d) + M_2 l_2 (l_1 C_2 + l_2) \\ m_2 d(l_1 C_2 + d) + M_2 l_2 (l_1 C_2 + l_2) & m_2 d^2 + M_2 l_2^2 \end{bmatrix}$$

- This is the **mass-inertia matrix**, it depends only on q_2 (inter-leg angle)
- Each element is of the form of an inertia (mass*length^2)
- Non-zero link inertias would add to the appropriate elements



DLM Modelling

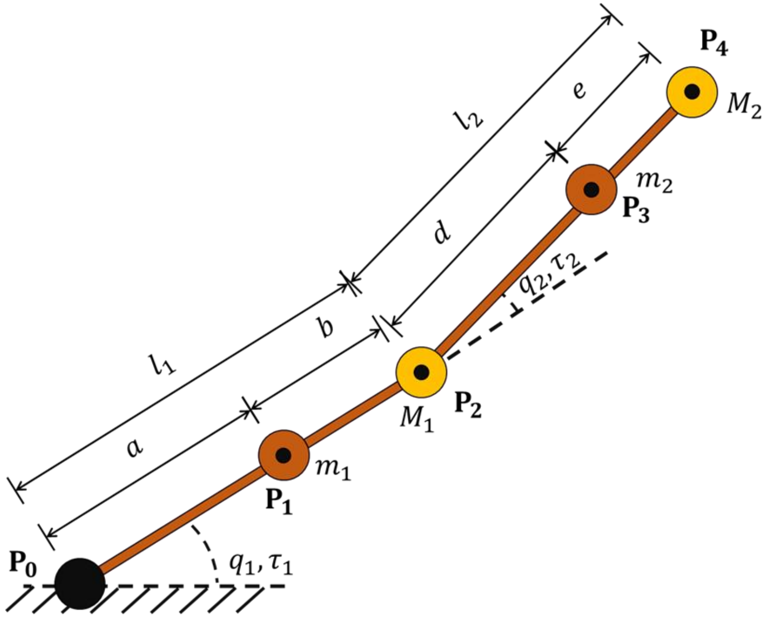
DLM Manipulator (Kinetic Energy)

- Replacing the Kinetic energy in the Euler-Lagrange equation we obtain

$$\frac{d}{dt}\left(\frac{\partial}{\partial \dot{\boldsymbol{q}}} KE(\boldsymbol{q},\dot{\boldsymbol{q}})\right) - \frac{\partial}{\partial \boldsymbol{q}} KE(\boldsymbol{q},\dot{\boldsymbol{q}}) + \frac{\partial}{\partial \boldsymbol{q}} PE(\boldsymbol{q}) = \tau$$

$$\frac{d}{dt}\left(\frac{\partial}{\partial \dot{\boldsymbol{q}}}\left(\frac{1}{2}\,\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\right)\right) - \frac{\partial}{\partial \boldsymbol{q}}\left(KE(\boldsymbol{q},\dot{\boldsymbol{q}})\right) + \frac{\partial}{\partial q} PE(\boldsymbol{q}) = \tau$$

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \frac{\partial}{\partial \boldsymbol{q}}\left(KE(\boldsymbol{q},\dot{\boldsymbol{q}})\right) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$



This gives us the **angular acceleration** term (multiplied by mass-inertia matrix, as well as one component of the centripetal / centrifugal rotating frames term which is **quadratic** in the **angular velocities**

DLM Modelling

DLM Manipulator (Kinetic Energy)

- Let

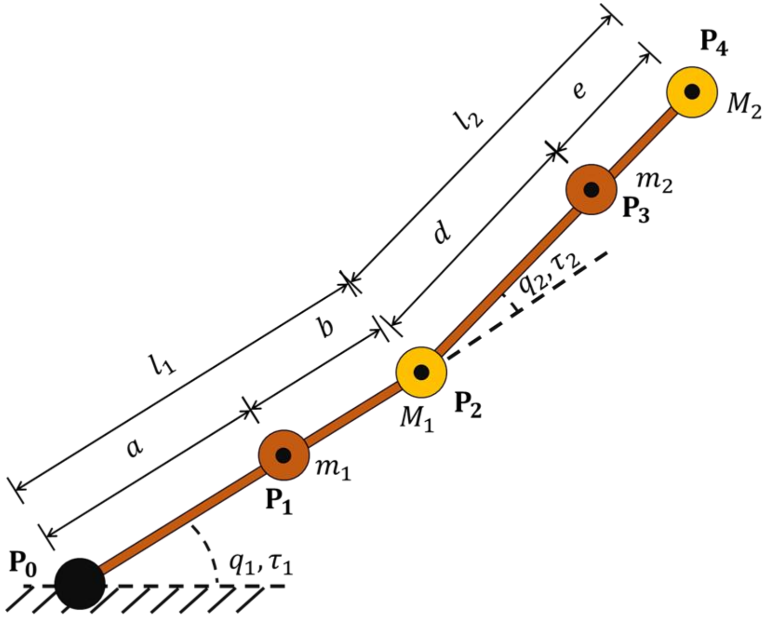
$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \dot{\boldsymbol{M}}(\boldsymbol{q})\dot{\boldsymbol{q}} - \frac{\partial}{\partial \boldsymbol{q}}\big(KE(\boldsymbol{q},\dot{\boldsymbol{q}})\big)$$

- Then

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

Where

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \begin{bmatrix} -2l_1S_2(m_2d + M_2l_2)\dot{q}_2 & -l_1S_2(m_2d + M_2l_2)\dot{q}_2 \\ -l_1S_2(m_2d + M_2l_2)\dot{q}_2 & 0 \end{bmatrix}$$



DLM Modelling

: BC C Q^Ÿ e\Qd_b¥AY^UdYS; ^UbW¦

- And

$$\frac{\partial}{\partial \boldsymbol{q}}\left(KE(\boldsymbol{q},\dot{\boldsymbol{q}})\right) = \begin{bmatrix} \frac{\partial}{\partial q_1}\left(KE(\boldsymbol{q},\dot{\boldsymbol{q}})\right) \\ \frac{\partial}{\partial q_2}\left(KE(\boldsymbol{q},\dot{\boldsymbol{q}})\right) \end{bmatrix} = \begin{bmatrix} 0 \\ -l_1S_2(m_2d+M_2l_2)\dot{q}_1^2 - l_1S_2(m_2d+M_2l_2)q_1q_2 \end{bmatrix}$$

- Putting everything together

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \begin{bmatrix} -2l_1S_2(m_2d+M_2l_2)\dot{q}_2 & -l_1S_2(m_2d+M_2l_2)\dot{q}_2 \\ -l_1S_2(m_2d+M_2l_2)\dot{q}_2 & 0 \end{bmatrix}\dot{\boldsymbol{q}} - \begin{bmatrix} 0 \\ -l_1S_2(m_2d+M_2l_2)\dot{q}_1^2 - l_1S_2(m_2d+M_2l_2)q_1q_2 \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \begin{bmatrix} -2l_1S_2(m_2d+M_2l_2)\dot{q}_2 & -l_1S_2(m_2d+M_2l_2)\dot{q}_2 \\ l_1S_2(m_2d+M_2l_2)\dot{q}_1 & 0 \end{bmatrix}\dot{\boldsymbol{q}}$$

DLM Modelling

: BC C Q^Y e\Qd_b¥C _TU\

- Then

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

- Where

$$M(\boldsymbol{q}) = \begin{bmatrix} m_1a^2 + M_1l_1^2 + m_2(l_1^2 + d^2 + 2l_1dC_2) + M_2(l_1^2 + l_2^2 + 2l_1l_2C_2) & m_2d(l_1C_2 + d) + M_2l_2(l_1C_2 + l_2) \\ m_2d(l_1C_2 + d) + M_2l_2(l_1C_2 + l_2) & m_2d^2 + M_2l_2^2 \end{bmatrix}$$

Mass (inertia) matrix (mass*length^2)

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} -2l_1S_2(m_2d + M_2l_2)\dot{q}_2 & -l_1S_2(m_2d + M_2l_2)\dot{q}_2 \\ l_1S_2(m_2d + M_2l_2)\dot{q}_1 & 0 \end{bmatrix}$$

Centrifugal / centripetal vector of rotational forces (mass*length^2*velocity^2)

$$\mathbf{g}(\mathbf{q}) = g\begin{bmatrix} m_1aC_1 + M_1l_1C_1 + m_2(l_1C_1 + dC_{12}) + M_2(l_1C_1 + l_2C_{12}) \\ C_{12}(m_2d + M_2l_2) \end{bmatrix}$$

Gravity vector. Torques required to counteract gravity (force*length)

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

Vector of applied (joint) **torques**. For the **passive** compass gait, this is zero

Conclusions

While a little complex, the equations of motion of the compass gait (double link, planar) robot can be derived using Euler-Lagrange. It is difficult to do this using a simple balance approach.

- The equations of motion contain the non-linear, 2nd order relationships between the applied joint torques (inputs) and the joint angles (accelerations) (outputs)

Interpreting & Simulating the Model

To gain some **insight** into the actual behaviour of the **swing phase dynamics**, we can

1) **Analysing** bits of the ODE,
2) **Simulating** the non-linear behaviour
3) **Linearized analysis** about operating points.

In this section, the analysis will be on the ODE's parts:

$$\underbrace{\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}}_{\text{Mass-Inertia matrix}} + \underbrace{\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}}_{\text{Rotational effects}} + \underbrace{\boldsymbol{g}(\boldsymbol{q})}_{\text{Gravity vector}} = \boldsymbol{\tau}$$

- We typically assume the other terms on the left hand size are zero and analyse each term independently
- **Simulate** the **autonomous behaviour**
  - Implement in ROS/Simulink/Matlab
  - Analyse autonomous response

Gravity Vector & Equilibrium Torque

Gravity Vector

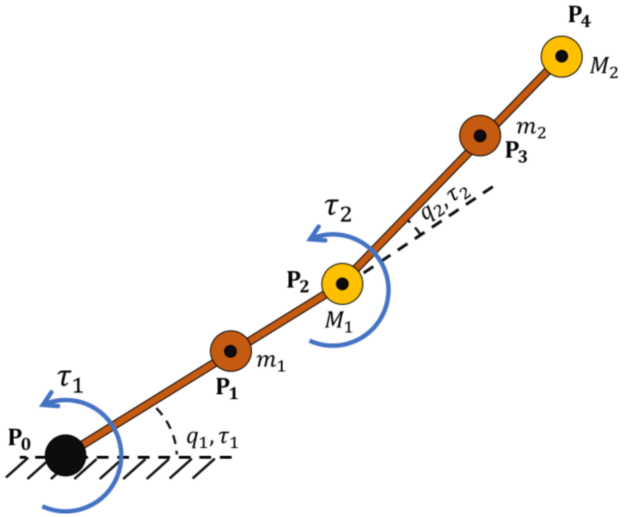
$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau}$$

To gain an insight into the gravity vector $\mathbf{g}(\mathbf{q})$, assume that the system is in equilibrium $\ddot{\mathbf{q}} = \dot{\mathbf{q}} = 0$

$$\boldsymbol{g}(\boldsymbol{q}) = g\begin{bmatrix} m_1 a C_1 + M_1 l_1 C_1 + m_2(l_1 C_1 + d C_{12}) + M_2(l_1 C_1 + l_2 C_{12}) \\ C_{12}(m_2 d + M_2 l_2) \end{bmatrix}$$

- It is termed the gravity vector.
- Represents the torque on each joint due to gravity acting on the point masses.

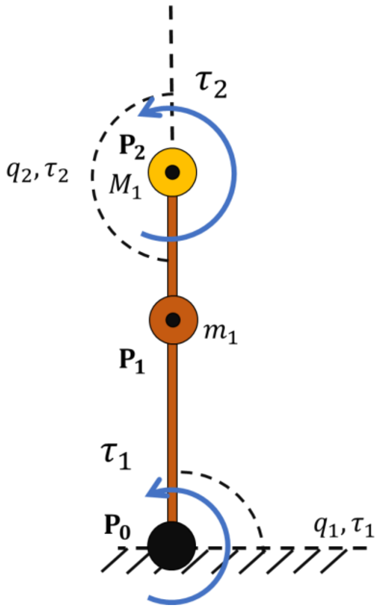
- It represents the **equilibrium torques** (acting at the joints) necessary to hold the "manipulator" at zero acceleration (velocity)
- A direct generalization of the same concept for the single link manipulator



Examples Equilibrium Torque

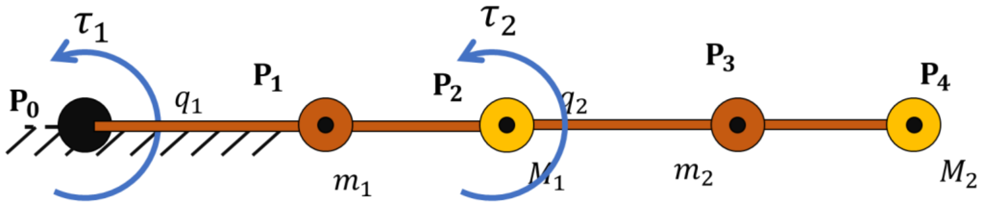
Examples

1. Vertical Upright ($q_1 = \frac{\pi}{2}, q_2 = \pi$): The equilibrium joint torques are both **zero** as the legs are in an unstable equilibrium configuration.



2. Max Static Torque Position ($q_1 = 0, q_2 = 0$): This represents a (static) posture. The calculated torques assume zero velocity which isn't a realistic part of the cycle, but the values for the exemplar manipulator are:

$$\boldsymbol{\tau} = \begin{bmatrix}\tau_1 \\ \tau_2\end{bmatrix} = g\begin{bmatrix} m_1 a + M_1 l_1 + m_2(l_1 + d) + M_2(l_1 + l_2) \\ (m_2 d + M_2 l_2)\end{bmatrix}$$

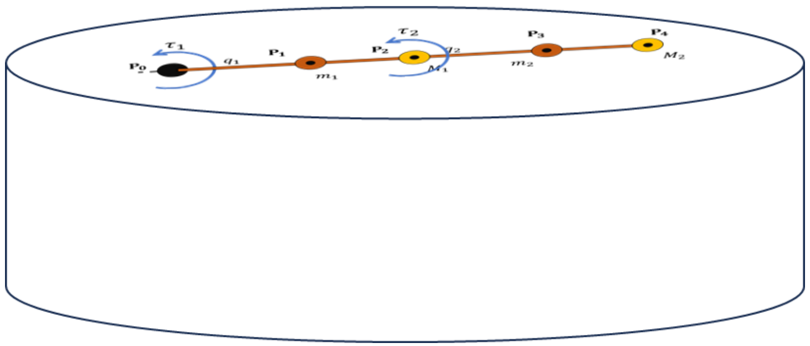


Interpreting the Mass Inertia Matrix

Mass Inertia Matrix

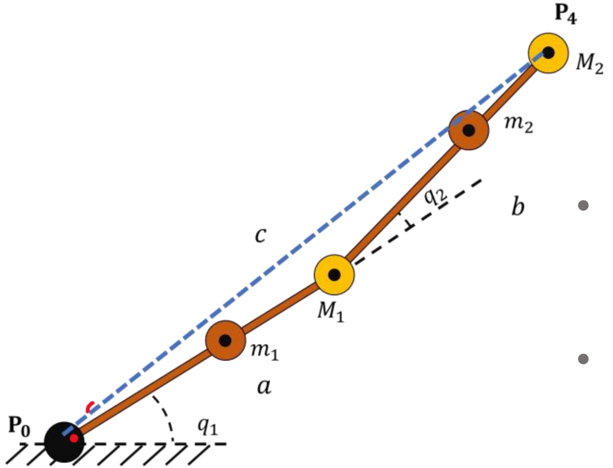
$$M(\boldsymbol{q}) = \begin{bmatrix} m_1a^2 + M_1l_1^2 + m_2(l_1^2 + d^2 + 2l_1dC_2) + M_2(l_1^2 + l_2^2 + 2l_1l_2C_2) & m_2d(l_1C_2 + d) + M_2l_2(l_1C_2 + l_2) \\ m_2d(l_1C_2 + d) + M_2l_2(l_1C_2 + l_2) & m_2d^2 + M_2l_2^2 \end{bmatrix}$$

- To understand the torque – angular acceleration map, assume we're in the horizontal plane ($g = 0$) (top of a table) and at zero speed, so $\mathbf{M}(q_2)\ddot{\mathbf{q}} = \boldsymbol{\tau}$



$$M(\boldsymbol{q}) = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

- $\tau_2 = m_{22}\ddot{q}_2$, where m_{22} is the inertia of a **single, rotating point mass**.
- $\tau_1 = m_{11}\ddot{q}_1$, where m_{11} is the inertia of **4 point masses** & depends on the q_2 using the law of cosines ($c^2 = a^2 + b^2 - 2ab\cos(\pi - q_2)$)



- Maximum when $q2 = 0$ becomes the inertia of the extended link
- Minimum when $q2 = \pi$

Interpreting the Mass Inertia Matrix 1

Mass Inertia Matrix

$M(\boldsymbol{q})$

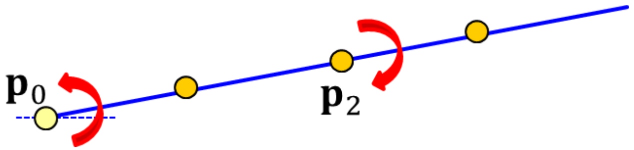
$$= \begin{bmatrix} m_1 a^2 + M_1 l_1^2 + m_2(l_1^2 + d^2 + 2l_1 d C_2) + M_2(l_1^2 + l_2^2 + 2l_1 l_2 C_2) & m_2 d(l_1 C_2 + d) + M_2 l_2(l_1 C_2 + l_2) \\ m_2 d(l_1 C_2 + d) + M_2 l_2(l_1 C_2 + l_2) & m_2 d^2 + M_2 l_2^2 \end{bmatrix}$$

- The symmetric, off-diagonal elements represent the interaction between the 1st torque, τ_1, and 2nd angular acceleration, $\ddot{q}_2$, and vice versa
- The interaction gain, $m_2 d(l_1 C_2 + d) + M_2 l_2(l_1 C_2 + l_2)$, can be positive or negative and depends on the relative joint angle q_2.

- positive gain when$m_2 d(l_1 C_2 + d) + M_2 l_2(l_1 C_2 + l_2) > 0$ & $\tau_1 > 0$

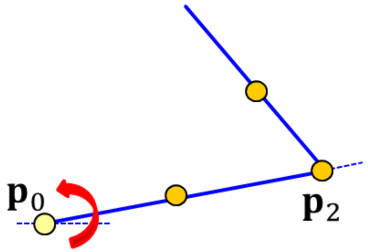
  $\tau_2 = 0 \rightarrow \ddot{q}_1 > 0 \rightarrow \ddot{q}_2 < 0$

  maximum when $q_2 = 0$



- Zero gain (independent) when$m_2 d(l_1 C_2 + d) + M_2 l_2(l_1 C_2 + l_2) = 0$ & $\tau_1 > 0$

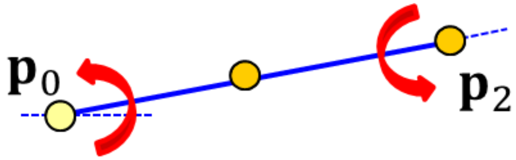
  $\tau_2 = 0 \rightarrow \ddot{q}_1 > 0 \rightarrow \ddot{q}_2 = 0$



- Negative gain when$m_2 d(l_1 C_2 + d) + M_2 l_2(l_1 C_2 + l_2) < 0$ & $\tau_1 > 0$

  $\tau_2 = 0 \rightarrow \ddot{q}_1 > 0 \rightarrow \ddot{q}_2 > 0$

  minimum when $q_2 = \pi$



Interpreting $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$

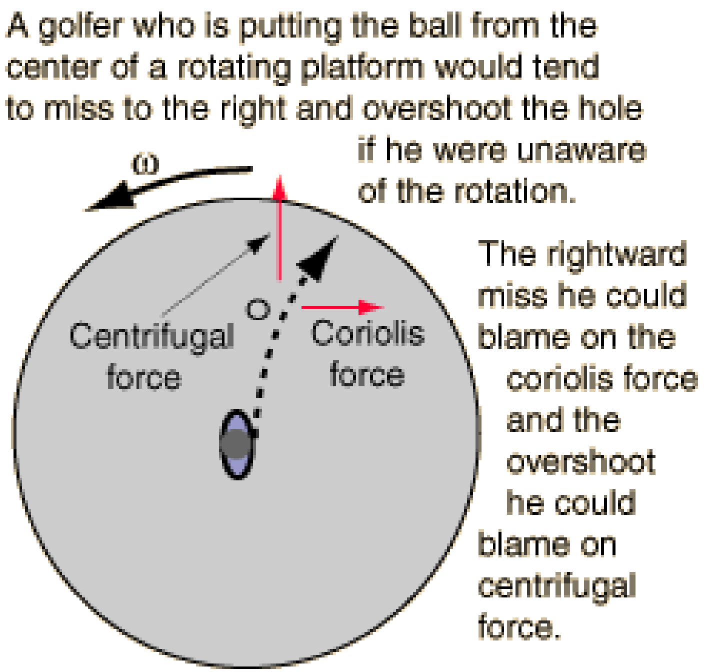
$$C(\boldsymbol{q},\dot{\boldsymbol{q}}) = \begin{bmatrix} -2l_1 S_2(m_2 d + M_2 l_2)\dot{q}_2 & -l_1 S_2(m_2 d + M_2 l_2)\dot{q}_2 \\ l_1 S_2(m_2 d + M_2 l_2)\dot{q}_1 & 0 \end{bmatrix}$$

- Always non-zero when there is more than 1 link
- **Centripetal** and **coriolis** forces are "effective forces" which are invoked to explain the behaviour of objects from a frame of reference which is rotating.
- Centripetal force: $F = mr\omega^2$

$$[\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}]_i = \boxed{\sum_j C_{ijj}\dot{q}_j^2} + \boxed{\sum_{\substack{j,k \\ j\neq k}} C_{ijk}\dot{q}_j\dot{q}_k}$$

Centripetal effects

Coriolis effects



Non-linear, State Space Swing Phase

The non-linear equations of motion for the swing phase can be re-written in state space (1^{st} order, vector) form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

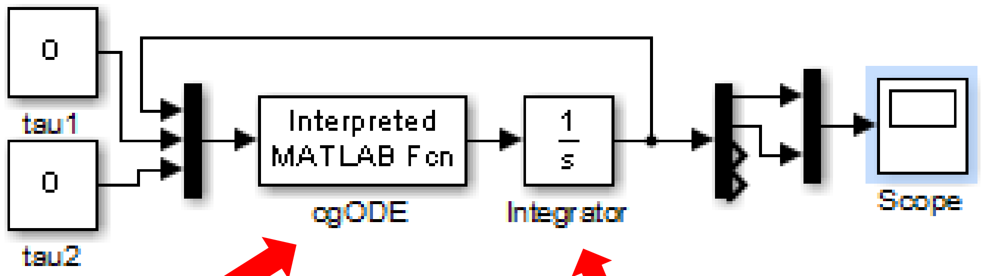
$$\frac{d}{dt}\begin{bmatrix}\mathbf{q} \\ \dot{\mathbf{q}}\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1}(\boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{g}(\mathbf{q}))\end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix}\mathbf{q} \\ \dot{\mathbf{q}}\end{bmatrix}, \quad \mathbf{u} = \boldsymbol{\tau}$$

Note that even for this very simple system (two links without motor models), neither the equations of motion nor the dynamic behaviour are trivial.

This 4^{th} order (4 states: angles and angular velocities) dynamic system, can be "easily" implemented in Matlab / Simulink by writing the ODE in an .m file and running in Matlab (`ode45()` function) or in Simulink.

We'll simulate the natural / autonomous behaviour where $\boldsymbol{\tau} = 0$

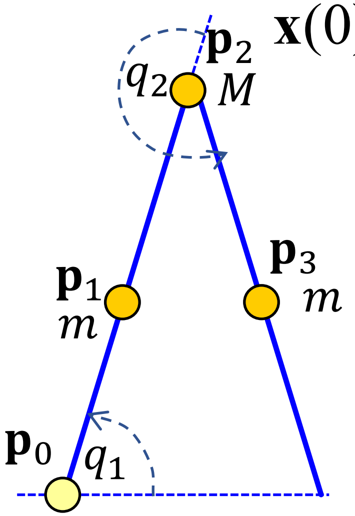
Swing Phase ODE Implementation



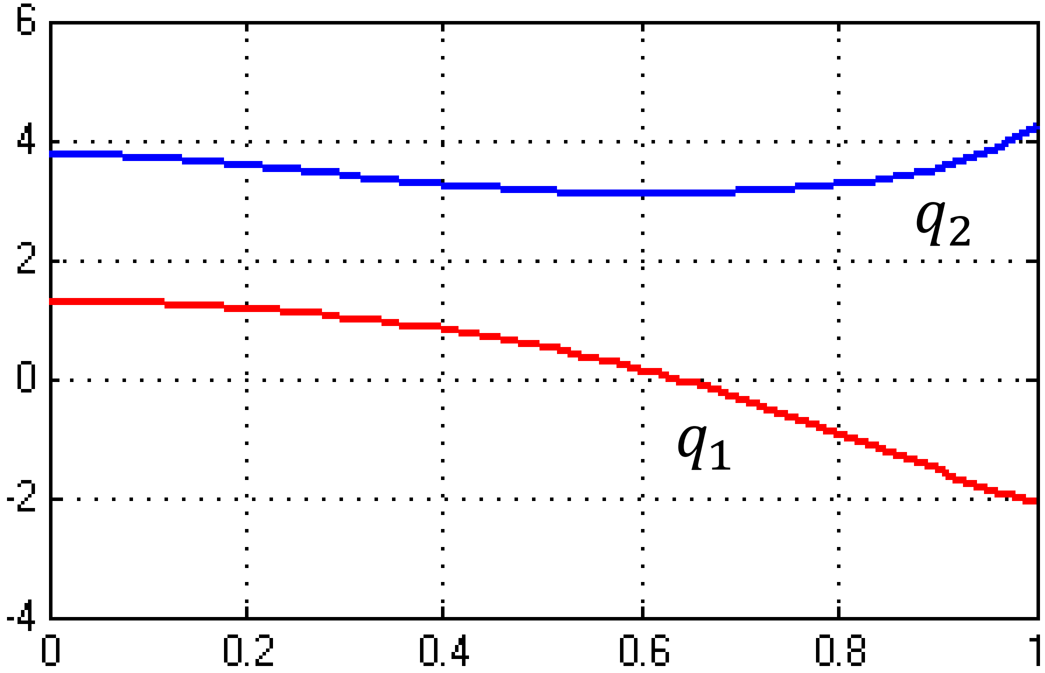

```
function dx = cgODE(x, u)
dx = zeros(4,1);
% ODE parameters
m = 5; M = 10; g = 9.8;
l = 1; a = 0.5; b = 0.5;
% Define ODE terms
Mm = [M*l^2+m*(l^2+a^2+b^2+2*b*l*cos(x(2))), m*b*(b+l*cos(x(2))); ...
    m*b*(b+l*cos(x(2))), m*b^2];
Cq = [-2*m*b*l*sin(x(2))*x(3)*x(4)-m*b*l*sin(x(2))*x(4)^2; ...
    m*b*l*sin(x(2))*x(3)^2];
gv = g*[(m*a+m*l+M*l)*cos(x(1))+m*b*cos(x(1)+x(2)); ...
    m*b*cos(x(1)+x(2))];
% Calculate xdot = f(x)
dx(1:2) = x(3:4);
dx(3:4) = inv(Mm)*(u-Cq-gv);
```

$$\mathbf{x}(0) = \begin{bmatrix} \pi/2 - 0.3 \\ \pi + 0.6 \\ 0 \\ 0 \end{bmatrix}$$

Initial Simulation Results

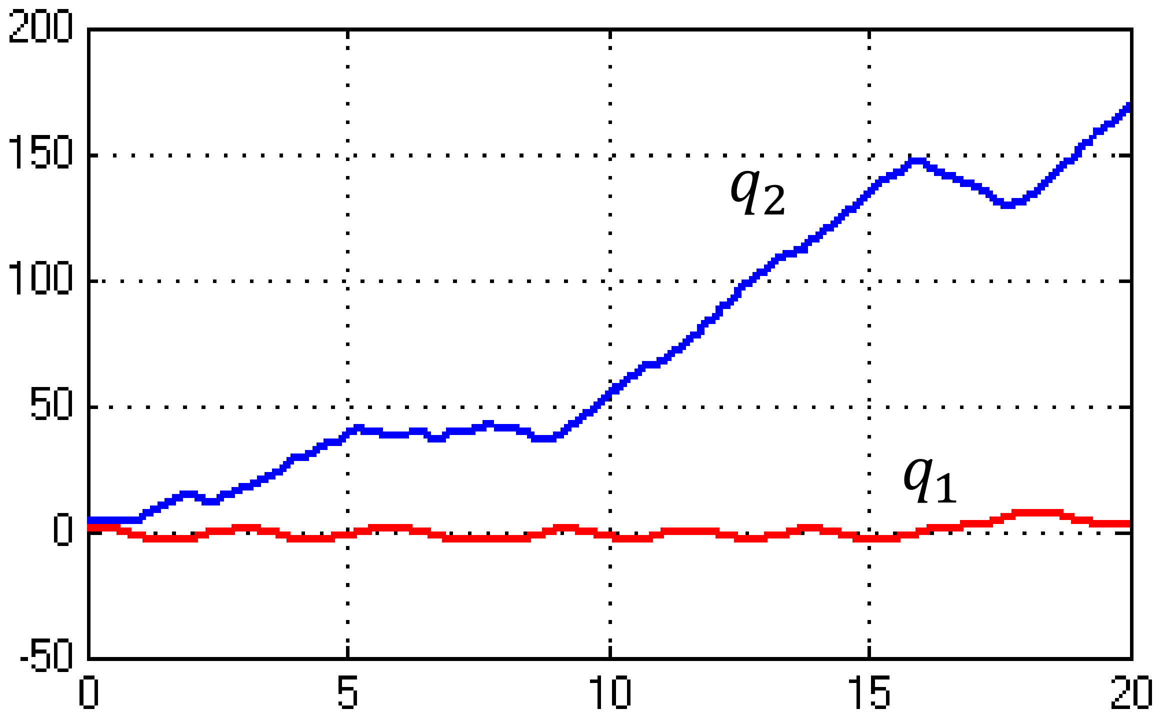


$$\mathbf{x}(0) = \begin{bmatrix} \pi/2 - 0.3 \\ \pi + 0.6 \\ 0 \\ 0 \end{bmatrix}$$



- The link 1 initially falls forward as q_1 decreases. In fact, it is rotating in a clockwise direction and passes the downwards vertical after about 0.8 s
- The link 2 is pulled in and when the stance leg is horizontal, the swing leg also becomes horizontal, before increasing q_2

3.5 Long Term Simulation Results



- The long term behaviour of the two link system is chaotic, it is extremely sensitive to the initial conditions
- We won't prove its chaotic, but the unstable behaviour (exponential divergence) coupled with the folding (rotating links) is characteristic of chaotic systems
- http://en.wikipedia.org/wiki/Double_pendulum (info + animation)

Conclusions

While the equations of motion appear quite complex, their interpretation isn't too hard as all the additive terms are regarded as torques

- The gravity vector, $\mathbf{g}(\mathbf{q})$, is easy to interpret as the effect of gravity on the joints (equilibrium torques)
- The elements of the mass – inertia matrix $\mathbf{M}(q_2)$ can be interpreted as "equivalent inertia" terms
  - The (positive) diagonal terms represent the direct $\tau_i \rightarrow q_i$ joint mapping
  - The off diagonal terms (can be negative as well as positive, or zero) reflect the interaction maps which reflect a joint angle's acceleration due to a torque applied at another joint
- The rotational torques $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ term is more difficult to easily interpret but it is worth noting that it would have been much more difficult to derive from 1st principles forces approach