

6ELEN018W - Applied Robotics
Lecture 7: Robot Dynamics - Motion upon
Forces - Part II

Dr Dimitris C. Dracopoulos

There is No Royal Road to Robotics!

King Ptolemy to Euclid (300 BC): After finding “*Elements*” seminal books too difficult to study: “*Is there were a quicker way to learn Geometry?*”

Euclid response to King Ptolemy: “*Sire, there is no Royal Road to Geometry*”

Last Lecture - Newtonian and Lagrangian Mechanics

- Calculate the dynamic equations of motion for a robot, subject to (generalised) forces, i.e. linear forces and angular forces (torques), one can use either Newtonian or Lagrangian mechanics.

Robot Manipulator Rigid Body Equations of Motion

Consider serial-link manipulator and the motor which actuates each joint j , $j \in \{0, \dots, N\}$.

- ▶ The inertia that the motor experiences is a function (depends) of the configuration of the outward links $j_{i+1}, j_{i+2}, \dots, j_N$.
- ▶ The equations of motion can be derived using Newton's second law and Euler's equation of rotational motion or the Lagrangian energy-based method.

The actuator forces (joint torques) \mathbf{Q} can be written as a set of coupled differential equations:

$$\mathbf{Q} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}}) + \mathbf{J}^T(\mathbf{q})\mathbf{w} \quad (1)$$

- ▶ \mathbf{q} are the joint coordinates (angles)
- ▶ $\dot{\mathbf{q}}$ are the joint velocities
- ▶ $\ddot{\mathbf{q}}$ are the joint accelerations
- ▶ \mathbf{g} is a term which represents the torque due to the gravity acting on the manipulator. This depends only on the configuration (joint angles \mathbf{q})

Robot Manipulator Rigid Body Equations of Motion (cont'd)

$$\mathbf{Q} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}}) + \mathbf{J}^T(\mathbf{q})\mathbf{w}$$

- ▶ \mathbf{M} is the inertia matrix and depends only on the configuration of the robot (joint angles \mathbf{q})
- ▶ \mathbf{C} is referred to as the Coriolis and centripetal term and this represents the gyroscopic and other forces that act on the robot joints due to the rotation of other robot joints.
- ▶ \mathbf{f} is the friction force
- ▶ $\mathbf{J}(\mathbf{q})$ is the manipulator Jacobian, and $\mathbf{w} \in \mathbb{R}^6$ is the wrench (i.e. forces and torques) applied at the end-effector.

⇒ This is the **inverse dynamics problem**:

Given the motion find the torques: $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \rightarrow \mathbf{Q}$

The Newton-Euler Recursive Formula

Solve the equations of motion for the robot serial-link manipulator.
How it works?

- ▶ Determine the translational and rotational velocity and acceleration for the centre of mass of each link.
 - ▶ Use Newton's second law for translational motion.
 - ▶ Use Euler's law for rotational motion.
- ▶ Start at the base of the robot and work outwards to:
 - ▶ Determine the translational and angular velocity of the centre of mass for each link in turn.
- ▶ Once we reach the end of the robot: start at the tip and work inwards:
 - ▶ Determine the force and moment each link exerts on the inboard link

Using the Robotics Toolbox for the Newton-Euler Recursive Formula

```
puma = models.DH.Puma560() # 6-joint robot
```

```
zero = np.zeros(6)
```

```
# print nominal configuration
```

```
print(puma.qn)
```

```
puma.plot(puma.qn)
```

```
Q = puma.rne(puma.qn, zero, zero)
```

The robot is not moving ($\mathbf{q} = 0$, $\dot{\mathbf{q}} = 0$), therefore these torques must be those required to hold the robot up against gravity.

Without gravity:

```
Q = puma.rne(puma.qn, zero, zero, gravity=[0, 0, 0])
```

Using the Robotics Toolbox for the Newton-Euler Recursive Formula

Consider now a case where the robot is moving, joint 1 has a velocity of 1 *rad/sec*. In the absence of gravity, the required joint torques are:

```
puma.rne(puma.qn, [1, 0, 0, 0, 0, 0], zero, gravity=[0, 0, 0])
```

The torque on joint 0 is that needed to overcome friction which always opposes the motion. The nonzero torques need to be exerted on the joints to oppose the gyroscopic torques that joint 0 motion is exerting on those joints.

Gravity and Payload

The equations of motion (1) of the serial link robot manipulator:

$$\mathbf{Q} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \underline{\mathbf{g}(\mathbf{q})} + \mathbf{f}(\dot{\mathbf{q}}) + \mathbf{J}^T(\mathbf{q})\mathbf{w}$$

- ▶ Gravity is the force that acts on the robot even if it's not moving.
- ▶ The torque that counteracts gravity and stops the arm from collapsing under its own weight.

Payload

A robot needs has to carry an object and place it somewhere else. This is the end-effector's payload.

- ▶ The last link in the chain of the robot has to hold the payload.
- ▶ This propagates down the chain towards the base of the robot.
- ▶ All joints of the robot need to help hold up the payload to stop it being pulled down by the force of gravity.

Effect of the payload:

- ▶ As the mass of the object (payload) increases:

⇒ *One joint will hit its torque limit it will become overloaded.
And that's the maximum payload that the robot can hold.*

- ▶ The maximum payload of the robot is a function of the torque capabilities of the motors but it is also a function of the configuration (angles) of the robot links.

Calculating the Gravity load (torques) Using the Robotics Toolbox

```
Q = puma.gravload(puma.qn)
```

```
# The default gravitational force in Earth  
puma.gravity
```

```
# Gravity in moon:  
print(puma.gravity / 6)
```

```
# Place the robot in moon  
puma.gravity = puma.gravity / 6
```

```
# In moon the torques required are reduced:  
  
print(puma.gravload(puma.qn))
```