AI Robotics

Learning Objectives

- Understand the Lagrangian and Euler-Newton formulation of dynamics
- Apply dynamic calculations to open-chain robots

Outline

Dynamics

- In previous cases dealing with kinematics we ignored how the forces and torques relate to the motion of the rigid bodies forming the robot.
- Here we introduce the dynamics of the robots and the associted equations of motion describing the system
- In general these form a set of second-order differential equations

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})$$

- $\theta \in \mathrm{R}^n$ are the joint variables, $\tau \in \mathrm{R}^n$ describe the joint forces and torques, $M(\theta) \in \mathrm{R}^{n \times n}$ is the mass matrix and $h(\theta, \dot{\theta})$ describe forces including centripetal, Coriolis, gravity and friction related forces.
- As for kinematics dynamics has both forward and inverse components

Dynamics

- The forward dyanmics problem is the determination of the robot acceleration $\ddot{\theta}$ given the joint configuration θ, joint velocities $\dot{\theta}$ and any joint forces and torques

$$\ddot{\theta} = M^{-1}(\theta)\left(\tau - h(\theta, \dot{\theta})\right)$$

- The inverse problem is the determination of the joint forces and torques for a given state and target acceleration

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})$$

Dynamics

- There are typically two formulations for the equations of motion
- The Lagrangian Formulation
  - Based on the energy (kinetic and potential) of the system
  - Provides closed form soltuions for the dynamics
  - Calculations become difficult with signifcant numbers of degrees of fredom
- Newton-Euler Formulation
  - Based on the balance of forces and torques
  - Solutions are numerical and efficient to calculate

Outline

1. Dynamics
   - Lagrangian Dynamics
   - Newton-Euler Dynamics

Lagrangian Formulation

- In order to work within the Lagrangian formulation a set of coordinates q need to be defined which describe the cofiguration of the system. These are called generalised coordinates
- Given these generalised coordinates the generalised forces f can be defined
- The Lagrangian $\mathcal{L}(q, \dot{q})$ is given by the kinetic energy and potential energy as

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{P}(q)$$

Kinetic Energy

- Kinetic Energy of a point with mass m and position $p(t)$ is given by

$$\mathcal{K} = \frac{1}{2} m \dot{p}^2$$

- For a rigid body we consider the body as a collection of point particles, each with different velcoties in the fixed frame and the kinetic energy is given by

$$\mathcal{K} = \frac{1}{2} \sum_i m_i \dot{p}_i^2$$

- With the knowledge of the Lagrangian $\mathcal{L}$ the equations of motion can be computed by

$$f = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\mathcal{L}}{\partial q}$$

Lagrangian Example Planar 2R

$$\tau = M(\theta)\ddot{\theta} + c(\theta, \dot{\theta}) + g(\theta)$$

$$M(\theta) = \begin{bmatrix} \mathfrak{m}_1 L_1^2 + \mathfrak{m}_2 \left(L_1^2 + 2L_1 L_2 \cos\theta_2 + L_2^2\right) & \mathfrak{m}_2 \left(L_1 L_2 \cos\theta_2 + L_2^2\right) \\ \mathfrak{m}_2 \left(L_1 L_2 \cos\theta_2 + L_2^2\right) & \mathfrak{m}_2 L_2^2 \end{bmatrix}$$

$$c(\theta, \dot{\theta}) = \begin{bmatrix} -\mathfrak{m}_2 L_1 L_2 \sin\theta_2 \left(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right) \\ \mathfrak{m}_2 L_1 L_2 \dot{\theta}_1^2 \sin\theta_2 \end{bmatrix}$$

$$g(\theta) = \begin{bmatrix} \left(\mathfrak{m}_1 + \mathfrak{m}_2\right) L_1 g \cos\theta_1 + \mathfrak{m}_2 g L_2 \cos\left(\theta_1 + \theta_2\right) \\ \mathfrak{m}_2 g L_2 \cos\left(\theta_1 + \theta_2\right) \end{bmatrix}$$

$$\begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -L_1\dot{\theta}_1^2 \\ -L_2\dot{\theta}_1^2 - L_2\dot{\theta}_2^2 \end{bmatrix}}_{\text{centripetal terms}} + \underbrace{\begin{bmatrix} 0 \\ -2L_2\dot{\theta}_1\dot{\theta}_2 \end{bmatrix}}_{\text{Coriolis terms}}$$

General Formulation

- We can describe a general formulation for a n-link open chain robot in the Lagrangian formulation.
- We will use the vector of joint values θ for the generalised coordinates
- For generalised forces τ if the joint is revolute τ will be a joint torque, for prismatic joints τ will correspond to a linear force.
- For rigid-body links the kinetic energy is always

$$\mathcal{K}(\theta, \dot{\theta}) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} m_{ij}(\theta) \dot{\theta}_i \dot{\theta}_j = \frac{1}{2} \dot{\theta}^{\mathrm{T}} M(\theta) \dot{\theta}$$

- Potential energies are given as

$$\mathcal{P}_\theta = \sum_{i=1}^{n} m_i g h_i(\theta)$$

Here the function $h_i(\theta)$ are the heights of the Center of mass of the link

- Recall the definition of the Lagrangian and generalised forces

$$\mathcal{L}(\theta, \dot{\theta}) = \mathcal{K}(\theta, \dot{\theta}) - \mathcal{P}(\theta)$$
$$\tau_i = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}} - \frac{\mathcal{L}}{\partial q}$$

- Using our earlier definition for the kinetic energy τ becomes

$$\tau_i = \sum_{j=1}^{n} m_{ij}(\theta)\ddot{\theta}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} \Gamma_{ijk}(\theta)\dot{\theta}_j\dot{\theta}_k + \frac{\partial \mathcal{P}}{\partial \theta_i}$$

- $\Gamma_{ijk}(\theta)$ are the Christoffel symbols of the first kind

$$\Gamma_{ijk}(\theta) = \frac{1}{2}\left(\frac{\partial M_{ij}}{\partial \theta_k} + \frac{\partial M_{ik}}{\partial \theta_j} - \frac{\partial M_{jk}}{\partial \theta_i}\right)$$

Mass Matrix

- The mass matrix $M(\theta)$ descibes the effective mass in different directions of acceleration
- Is a symmetric positive definite matrix

Classical rigid body dynamics

- Consider a generic right body made of many connected point masses. Each point mass i has a mass $\mathbf{m}_i$ and the total mass $\mathbf{m} = \sum_i m_i$.
- Each point mass is at a position $r_i = (x_i, y_i, z_i)$ in the body frame f located at the center of mass
- If the rigid body is moving along a body twist $\mathcal{V}_b$ the velocity of a sample point mass is given by

$$
\begin{aligned}
\dot{p}_i &= v_b + \omega \times p_i \\
\ddot{p}_i &= \dot{v}_b + \frac{d}{dt}\omega_b \times p_i + \omega_b \times \frac{d}{dt} p_t \\
&= \dot{v}_b + \dot{\omega}_b \times p_i \times (v_b + \omega_b \times p_i)
\end{aligned}
$$

- Utilisng our skew-symmetric notation in place of cross-products

$$\ddot{p}_i = \dot{v}_b + [\dot{\omega}_b]\, r_i + [\omega_b]\, v_b + [\omega_b]^2\, r_i$$

- From Newtons second law

$$\begin{aligned} f_i &= \mathfrak{m}_i \left(\dot{v}_b + [\dot{\omega}_b]\, r_i + [\omega_b]\, v_b + [\omega_b]^2\, r_i \right) \\ m_i &= [r_i] f_i \end{aligned}$$

- The expresions for the forces and moments in the determine wrench can be simplified in the center of mass frame

$$
\begin{aligned}
f_b &= \sum_i \mathfrak{m}_i \left(\dot{v}_b + [\dot{\omega}_b]\, r_i + [\omega_b]\, v_b + [\omega_b]^2\, r_i \right) \\
&= \sum_i \mathfrak{m}_i \left(\dot{v}_b + [\omega_b]\, v_b \right) \\
&= \mathrm{m} \left(\dot{v}_b + [\omega_b]\, v_b \right)
\end{aligned}
$$

$$
\begin{aligned}
m_b &= \sum_i \mathfrak{m}_i [r_i] \left(\dot{v}_b + [\dot{\omega}_b]\, r_i + [\omega_b]\, v_b + [\omega_b]^2\, r_i \right) \\
&= \sum_i \mathfrak{m}_i \left(- [r_i]^2\, \dot{\omega}_b - [r_i]^{\mathrm{T}} [\omega_b]^{\mathrm{T}} [r_i]\, \omega_b \right) \\
&= \sum_i \mathfrak{m}_i \left(- [r_i]^2\, \dot{\omega}_b - [\omega_b] [r_i]^2\, \omega_b \right) \\
&= \left(- \sum \mathfrak{m}_i\, [r_i]^2 \right) \dot{\omega}_b + [\omega_b] \left(- \sum \mathfrak{m}_i\, [r_i]^2 \right)
\end{aligned}
$$

Rotational Inertial Matrix

$$\mathcal{I}_b = \begin{bmatrix} \sum \mathfrak{m}_i \left(y_i^2 + z_i^2\right) & -\sum \mathfrak{m}_i x_i y_i & -\sum \mathfrak{m}_i x_i z_i \\ -\sum_{\mathfrak{m}_i} x_i y_i & \sum \mathfrak{m}_i \left(x_i^2 + z_i^2\right) & -\sum_{\mathfrak{m}_i} y_i z_i \\ -\sum_{\mathfrak{m}_i} x_i z_i & -\sum \mathfrak{m}_i y_i z_i & \sum \mathfrak{m}_i \left(x_i^2 + y_i^2\right) \end{bmatrix}$$

$$= \begin{bmatrix} \mathcal{I}_{xx} & \mathcal{I}_{xy} & \mathcal{I}_{xz} \\ \mathcal{I}_{xy} & \mathcal{I}_{yy} & \mathcal{I}_{yz} \\ \mathcal{I}_{xz} & \mathcal{I}_{yz} & \mathcal{I}_{zz} \end{bmatrix}$$

Rotational Inertial Matrix

- If we replace the point masses with a density ρ we replace the sums with integrals

$$\mathcal{I}_{xx} = \int_{\mathcal{B}} \left(y^2 + z^2\right) \rho(x, y, z) dV$$

$$\mathcal{I}_{yy} = \int_{\mathcal{B}} \left(x^2 + z^2\right) \rho(x, y, z) dV$$

$$\mathcal{I}_{zz} = \int_{\mathcal{B}} \left(x^2 + y^2\right) \rho(x, y, z) dV$$

$$\mathcal{I}_{xy} = -\int_{\mathcal{B}} xy \rho(x, y, z) dV$$

$$\mathcal{I}_{xz} = -\int_{\mathcal{B}} xz \rho(x, y, z) dV$$

$$\mathcal{I}_{yz} = -\int_{\mathcal{B}} yz \rho(x, y, z) dV$$

Twist and Wrench Dynamics

- The combination of rotational and linear dynamics can be written in matrix form as

$$\begin{bmatrix} m_b \\ f_b \end{bmatrix} = \begin{bmatrix} \mathcal{I}_b & 0 \\ 0 & \mathrm{m}I \end{bmatrix} \begin{bmatrix} \dot{\omega}_b \\ \dot{v}_b \end{bmatrix} + \begin{bmatrix} [\, \omega_b \,] & 0 \\ 0 & [\omega_b] \end{bmatrix} \begin{bmatrix} \mathcal{I}_b & 0 \\ 0 & \mathrm{m}I \end{bmatrix} \begin{bmatrix} \omega_b \\ v_b \end{bmatrix}$$

- We can see the familar forms of the body twist $\mathcal{V}_b$ and body wrench $\mathcal{F}_b$ in the above matrices

$$\mathcal{V}_b = \begin{bmatrix} \omega_b \\ v_b \end{bmatrix}, \begin{bmatrix} m_b \\ f_b \end{bmatrix}$$

- We can define the spatial inertia matrix as

$$\mathcal{G}_b = \begin{bmatrix} \mathcal{I}_b & 0 \\ 0 & mI \end{bmatrix}$$

- The kinetic energy of a rigid body can then be written purely in terms of body twists and the spatial inertial matrix

$$\mathcal{K} = \frac{1}{2}\omega_b^{\mathrm{T}}\mathcal{I}_b\omega_b + \frac{1}{2}\mathfrak{m}v_b^{\mathrm{T}}v_b = \frac{1}{2}\mathcal{V}_b^{\mathrm{T}}\mathcal{G}_b\mathcal{V}_b$$

- We can define the spatial momentum $\mathcal{P}_b$ as

$$\mathcal{P}_b = \left[\begin{array}{c} \mathcal{I}_b\omega_b \\ \mathfrak{m}v_b \end{array}\right] = \left[\begin{array}{cc} \mathcal{I}_b & 0 \\ 0 & \mathfrak{m}I \end{array}\right]\left[\begin{array}{c} \omega_b \\ v_b \end{array}\right] = \mathcal{G}_b\mathcal{V}_b$$

$$\begin{aligned} \mathcal{F}_b &= \mathcal{G}_b\dot{\mathcal{V}}_b - \mathrm{ad}_{\mathcal{V}_b}^{\mathrm{T}}\left(\mathcal{P}_b\right) \\ &= \mathcal{G}_b\dot{\mathcal{V}}_b - \left[\mathrm{ad}_{\mathcal{V}_b}\right]^{\mathrm{T}}\mathcal{G}_b\mathcal{V}_b \end{aligned}$$

Lie Brackets

- A Lie Bracket is a generalisation of the cross-product of two vectors applied to twists
- Given twists $\mathcal{V}_1 = (\omega_1, v_1)$ and $\mathcal{V}_2 = (\omega_2, v_2)$ the Lie bracket is

$$[ad_{\mathcal{V}_1}]\mathcal{V}_2 = \begin{bmatrix} [\omega_1] & 0 \\ [v_1] & [\omega_1] \end{bmatrix} \begin{bmatrix} \omega_2 \\ v_2 \end{bmatrix}$$

Changing frames

- Generally dynamics are calculated in the center of mass frame. However if alternative frames are desired it is possible to tranform the reference frame. Starting with the constraint that kinetic energy is indepented of reference frame

$$
\begin{aligned}
\frac{1}{2}\mathcal{V}_a^{\mathrm{T}}\mathcal{G}_a\mathcal{V}_a &= \frac{1}{2}\mathcal{V}_b^{\mathrm{T}}\mathcal{G}_b\mathcal{V}_b \\
&= \frac{1}{2}\left(\left[\mathrm{Ad}_{T_{ba}}\right]\mathcal{V}_a\right)^{\mathrm{T}}\mathcal{G}_b\left[\mathrm{Ad}_{T_{ba}}\right]\mathcal{V}_a \\
&= \frac{1}{\rho}\mathcal{V}_a^{\mathrm{T}}\left[\mathrm{Ad}_{T_{ba}}\right]^{\mathrm{T}}\mathcal{G}_b\left[\mathrm{Ad}_{T_{ba}}\right]\mathcal{V}_a
\end{aligned}
$$

Outline

- For a point mass m and coordinate $p(t)$, Newton's second law states $f = ma = m\ddot{p}(t)$
- For a rigid body with an associated rotational inertia matrix $\mathcal{I}_b$ and body twist $\mathcal{V}_b$ the resulting moments and forces are given by

$$
\begin{aligned}
m_b &= \mathcal{I}_b \dot{\omega}_b + \omega_b \times \mathcal{I}_b \omega_b \\
f_b &= \mathbf{m} \dot{v}_b + \omega_b \times \mathbf{m} v_b
\end{aligned}
$$

Inverse Dynamics

- For an n-link open chain, a reference frame i is attached to each link's center of mass
- Begining with the base frame as $n = 0$, the end effector frame is given by reference frame $n + 1$
- We compute the transformation matrix M_{ij} for joint j in frame i in the home position
- T_{ij} denotes the transformation of joint j in frame i at the joint angle θ

Newton-Euler Inverse Dynamics Algorithm

- Recall the goal is to calculate the right hand side of $\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta})$
- The Newton-Euler method consists of two stages:
  - The first stage is the forward iteration which obtains the configuration, twists and acceleration of each link starting from $n = 1$
  - The second stage is the backward iteration which determines the forces and torques starting from joint n

- Input
  - $\mathcal{V}_0$ and $\dot{\mathcal{V}}_0 = (\dot{\omega}_0, \dot{v}_0) = (0, -g)$
  - $\mathcal{F}_{n+1}$ is a known external wrench applied by the environment
- Forward Iteration
  - Given $\theta, \dot{\theta}, \ddot{\theta}$ for links $i = 1 \dots n$

$$T_{i,i-1}(\theta_i) = e^{-[\mathcal{A}_\rangle]\theta_i} M_{i,i-1}$$

$$\mathcal{V}_i = \left[Ad_{T_{i,i-1}}\right] \mathcal{V}_{i-1} + \mathcal{A}_i \dot{\theta}_i$$

$$\dot{\mathcal{V}}_i = \left[Ad_{T_{i,i-1}}\right] \dot{\mathcal{V}}_{i-1} + [Ad_{v_i}]\, \mathcal{A}_i \dot{\theta}_i + \mathcal{A}_i \ddot{\theta}_i$$

- Backwards Iteration

$$\mathcal{F}_i = \left[\mathrm{Ad}_{T_{i+1,i}}\right]^T \mathcal{F}_{i+1} + \mathcal{G}_i \dot{\mathcal{V}}_i - [\mathrm{ad} v_i]^T (\mathcal{G}_i \mathcal{V}_i)$$

$$\tau_i = \mathcal{F}_i^T \mathcal{A}_i$$

Open Chain Forward Dynamics

- For the forward dynamics problem we solve $M(\theta)\ddot{\theta} = \tau(t) - h(\theta, \dot{\theta}) - J^T(\theta)\mathcal{F}_{tip}$
- $h(\theta, \dot{\theta})$ can be calculated by solving the inverse dynamics while setting $\ddot{\theta} = 0$ and $\mathcal{F}_{tip} = 0$
- The Jacobian can be computed simply as we have seen before

- Inverse dynamics can be determined to give $\tau = \textit{InverseDynamics}(\theta, \dot{\theta}, \ddot{\theta}, \mathcal{F}_{tip})$
- By setting $\mathcal{F}_{tip} = 0$ and $\ddot{\theta} = 0$ $\tau = h(\theta, \dot{\theta})$
- So $h(\theta, \dot{\theta}) = \textit{InverseDynamics}(\theta, \dot{\theta}, 0, 0)$
- Addtionally by setting $\dot{\theta} = 0$ then $\tau = M(\theta)\ddot{\theta}$
- Each column of the mass matrix $M(\theta)$ can be found by $M_{:,j} = \textit{InverseDynamics}(\theta, 0, \ddot{\theta}_j, 0)$

Euler Integration for Forward Dynamics

- Inputs: Initial conditions θ and $\dot{\theta}$, input torques $\tau(t)$ and any end-effector wrenches $\mathcal{F}_{tip}(t)$
- Itegration timestep is given by $\delta t = duration/N$
- For $k = 0$ to $N - 1$
  - $\ddot{\theta} = ForwardDynamics(\theta(k), \dot{\theta}(k), \tau(k\delta t), \mathcal{F}_{tip}(k\delta t))$
  - $\theta(k+1) = \theta(k) + \dot{\theta}(k)\delta t$
  - $\dot{\theta}(k+1) = \dot{\theta}(k) + \ddot{\theta}(k)\delta t$