MECEE 4602: Introduction to Robotics, Fall 2019

Dynamics

The Jacobian-transpose method discussed in previous lectures for converting joint torques into an end-effector wrench applies when the manipulator is as static-equilibrium: the external wrench applied by the environment is balanced by joint torques. What if the manipulator is not at equilibrium though? In that case, since joint torques are not perfectly balanced by the environment, the manipulator will move.

In this lecture, we look at the relationship between joint torques and the resulting motion of the manipulator. We try to take into account all phenomena that influence this relationship, such as manipulator mass, inertia, gravity, external forces, friction, etc. Still, overall, we are looking to understand how joint torques $\boldsymbol{\lambda}$, applied when the robot is in a given state described by $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$, produce acceleration, and hence movement:

$$\ddot{\boldsymbol{q}} = f(\boldsymbol{\lambda}, \boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{1}$$

There are two main ways of studying this relationship: a Lagrangian approach, and a Newtonian approach. We will look at both in this lecture.

1 Lagrangian Mechanics Approach

There are two key relationships that are the basis for the Lagrangian approach to studying a dynamic system. The first one defines the Lagrangian L of the system as the difference between kinetic energy K and potential energy P:

$$L = K - P \tag{2}$$

In general, the state of a dynamic system is described by the position and velocity of the bodies in a Cartesian reference frame. However, a key advantage of the Lagrangian formulation is that it is not tied to a particular coordinate system: we can choose to express K and P as functions of **any state variables that define our system**. In particular, for robotic manipulators, we can choose to use joint values and velocities as our variables, so we can think of the Lagrangian as:

$$L = K(\boldsymbol{q}, \dot{\boldsymbol{q}}) - P(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{3}$$

In this case, the Lagrange equations tell us that:

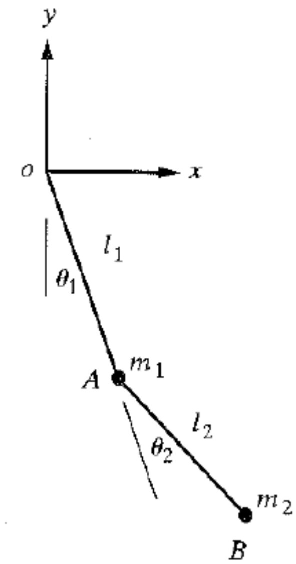
$$\lambda_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \tag{4}$$

where λ_i is the torque (or force for prismatic joints) acting on joint i.

This approach provides us with one way for obtaining the relationship between joint forces and movement. Let's apply it to a simple case and see what we get.

1.1 Example: 2-link manipulator, concentrated mass

Consider the case of a 2-link manipulator, where we assume the entire mass of each link is concentrated at the far end of the link (image and example from Niku, "Introduction to Robotics").



The kinetic energy of link 1 is straightforward:

$$K_1 = \frac{1}{2}mv_1^2 = \frac{1}{2}m_1 l_1^2 \dot{q}_1^2 \tag{5}$$

To compute the velocity of link 2, let's compute the position of it's far (distal) end:

$$\begin{aligned} x_2 &= l_1 S_1 + l_2 S_{12} && (6) \\ y_2 &= -l_1 C_1 - l_2 C12 && (7) \end{aligned}$$

$$\begin{aligned} \dot{x}_2 &= l_1 C_1 \dot{q}_1 + l_2 C_{12}(\dot{q}_1 + \dot{q}_2) && (8) \\ \dot{y}_2 &= l_1 S_1 \dot{q}_1 + l_2 S_{12}(\dot{q}_1 + \dot{q}_2) && (9) \end{aligned}$$

$$\begin{aligned} v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 &= l_1^2 C_1^2 \dot{q}_1^2 + l_2^2 C_{12}^2 (\dot{q}_1 + \dot{q}_2)^2 + 2 l_1 l_2 C_1 C_{12} (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) + \\ &+ l_1^2 S_1^2 \dot{q}_1^2 + l_2^2 S_{12}^2 (\dot{q}_1 + \dot{q}_2)^2 + 2 l_1 l_2 S_1 S_{12} (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) && (10) \\ &= l_1^2 \dot{q}_1^2 + l_2^2 (\dot{q}_1 + \dot{q}_2)^2 + 2 l_1 l_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) C_2 && (11) \end{aligned}$$

The potential energy of each link is:

$$\begin{aligned} P_1 &= -g m_1 l_1 C_1 && (12) \\ P_2 &= -g m_2 (l_1 C_1 + l_2 C_{12}) && (13) \end{aligned}$$

We can now compute the Lagrangian as:

$$\begin{aligned} L &= K - P = (K_1 + K_2) - (P_1 + P_2) = && (14) \\ &= \frac{1}{2}(m_1 + m_2) l_1^2 \dot{q}_1^2 + \frac{1}{2} m_2 l_2^2 (\dot{q}_1 + \dot{q}_2)^2 + m_2 l_1 l_2 C_2 (\dot{q}_1^2 + \dot{q}_1 \dot{q}_2) + \\ &+ g m_1 l_1 C_1 + g m_2 (l_1 C_1 + l_2 C_{12}) && (15) \end{aligned}$$

We then have:

$$\frac{\partial L}{\partial q_1} = -g(m_1+m_2)l_1 S_1 - g m_2 l_2 S_{12} \tag{16}$$

$$\frac{\partial L}{\partial \dot{q}_1} = (m_1+m_2)l_1^2\dot{q}_1 + m_2 l_2^2(\dot{q}_1+\dot{q}_2) + m_2 l_1 l_2 C_2(2\dot{q}_1+\dot{q}_2) \tag{17}$$

$$\tag{18}$$

$$\begin{aligned}\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} &= (m_1+m_2)l_1^2\ddot{q}_1 + m_2 l_2^2(\ddot{q}_1+\ddot{q}_2) - m_2 l_1 l_2 S_2(2\dot{q}_1\dot{q}_2+\dot{q}_2^2) + m_2 l_1 l_2 C_2(2\ddot{q}_1+\ddot{q}_2) = \\ &= [(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 C_2]\ddot{q}_1 + (m_2 l_2^2 + m_2 l_1 l_2 C_2)\ddot{q}_2 - \\ &- 2m_2 l_1 S_2\dot{q}_1\dot{q}_2 - 2m_2 l_1 S_2 \dot{q}_2^2\end{aligned} \tag{19}$$

And thus:

$$\begin{aligned}\lambda_1 &= \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} = \\ &= [(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 C_2]\ddot{q}_1 + (m_2 l_2^2 + m_2 l_1 l_2 C_2)\ddot{q}_2 - \\ &- 2m_2 l_1 S_2\dot{q}_1\dot{q}_2 - 2m_2 l_1 S_2 \dot{q}_2^2 + \\ &+ g(m_1+m_2)l_1 S_1 + g m_2 l_2 S_{12}\end{aligned} \tag{20}$$

Similarly, we can obtain:

$$\begin{aligned}\lambda_2 &= \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} = \\ &= (m_2 l_2^2 + m_2 l_1 l_2 C_2)\ddot{q}_1 + m_2 l_2^2\ddot{q}_2 + m_2 l_1 l_2 S_2 \dot{q}_1^2 + g m_2 l_2 S_{12}\end{aligned} \tag{21}$$

Putting together the result in matrix form, we obtain:

$$\begin{aligned}\begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix} &= \begin{bmatrix}(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 C_2 & m_2 l_2^2 + m_2 l_1 l_2 C_2\\ m_2 l_2^2 + m_2 l_1 l_2 C_2 & m_2 l_2^2\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} + \\ &+ \begin{bmatrix}0 & -m_2 l_1 l_2 S_2\\ m_2 l_1 l_2 S_2 & 0\end{bmatrix}\begin{bmatrix}\dot{q}_1^2\\ \dot{q}_2^2\end{bmatrix} + \\ &+ \begin{bmatrix}-m_2 l_1 l_2 S_2 & -m_2 l_1 l_2 S_2\\ 0 & 0\end{bmatrix}\begin{bmatrix}\dot{q}_1\dot{q}_2\\ \dot{q}_2\dot{q}_1\end{bmatrix} + \\ &+ \begin{bmatrix}g(m_1+m_2)l_1 S_1 + g m_2 l_2 S_{12}\\ g m_2 l_2 S_{12}\end{bmatrix}\end{aligned} \tag{22}$$

Or, in a slightly different form:

$$
\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} (m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 C_2 & m_2 l_2^2 + m_2 l_1 l_2 C_2 \\ m_2 l_2^2 + m_2 l_1 l_2 C_2 & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} +
$$

$$
+ \begin{bmatrix} -m_2 l_1 l_2 S_2 \dot{q}_2 & -m_2 l_1 l_2 S_2 (\dot{q}_1 + \dot{q}_2) \\ m_2 l_1 l_2 S_2 \dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} +
$$

$$
+ \begin{bmatrix} g(m_1+m_2) l_1 S_1 + g m_2 l_2 S_{12} \\ g m_2 l_2 S_{12} \end{bmatrix} \tag{23}
$$

We can now take a closer look at the terms that appear in (22) and (23):

- terms that get multiplied by $\ddot{q}_i$ represent **inertial forces**, or the contribution of link inertia to joint forces. Note that acceleration at one joint can create inertial forces at other joints!
- terms that get multiplied by $\dot{q}_i^2$ represent **centripetal forces**. Again, velocity at one joint can create centripetal forces at different joints.
- terms that get multiplied by $\dot{q}_i \dot{q}_j$ represent **Coriolis forces**. Again, velocity at one joint can create Coriolis forces at different joints.
- free terms represent other (usually external) forces acting on the system. In this particular case, they contain the effect of **gravity**.

Equation (23) is often summarized as:

$$
\boldsymbol{\lambda} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) \tag{24}
$$

where $\boldsymbol{M}$ is called the inertia matrix (or sometimes the mass matrix), $\boldsymbol{C}$ is the Coriolis and centripetal matrix, and $\boldsymbol{G}$ is the gravity load matrix. Additional free terms that can appear in (24) are a joint friction component $\boldsymbol{F}(\dot{\boldsymbol{q}})$, or the contribution of other external forces $\boldsymbol{\tau}$ (such as somebody pushing on a link), converted to joint space as $\boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\tau}$.

$$
\boldsymbol{\lambda} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}(\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\tau} \tag{25}
$$

1.2 Forward and inverse dynamics

The above relationships are mainly used in two ways:

- **forward (or direct) dynamics**: given joint torques $\boldsymbol{\lambda}$ and the state of the robot $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, compute the resulting accelerations $\ddot{\boldsymbol{q}}$. These can then be integrated numerically to obtain the robot's velocity and trajectory over time. Note that forward dynamics essentially requires solving a linear system:

$$
\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{q})[\boldsymbol{\lambda} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{F}(\dot{\boldsymbol{q}}) - \boldsymbol{G}(\boldsymbol{q}) - \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\tau}] \tag{26}
$$

- **inverse dynamics**: given the state of the robot ($\boldsymbol{q}$, $\dot{\boldsymbol{q}}$) compute the joint torques $\boldsymbol{\lambda}$ that achieve a desired acceleration $\ddot{\boldsymbol{q}}$. This is simply a matter of applying (25).

Note that, unlike the case of kinematics, inverse dynamics is computationally less expensive than forward dynamics.

1.3 Example poses

In all the following examples, we assume $m_1 = m_2 = 1g$, $l_1 = l_2 = 1m$.

Pose 1. As a first example, consider a stationary arm held horizontal against gravity: $\boldsymbol{q} = [\pi/2, 0]^T$, $\dot{\boldsymbol{q}} = [0, 0]^T$.

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} 29.4 \\ 9.8 \end{bmatrix} \tag{27}$$

If we set $\ddot{\boldsymbol{q}} = [0, 0]^T$, inverse dynamics simply gives us the joint torques we must apply to keep the robot stationary against gravity.

Let's assume we set λ_2 to twice the anti-gravity value, while keeping λ_1 the same. We get:

$$\begin{bmatrix} 29.4 \\ 19.6 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} 29.4 \\ 9.8 \end{bmatrix} \tag{28}$$

If we perform forward dynamics to compute joint accelerations, we get $\dot{\boldsymbol{q}} = [-19.6, 49.0]^T$. Joint 2 is accelerating, as expected, but, perhaps unexpectedly, joint 1 will move in the opposite direction due to inertia.

If we want just joint 2 to accelerate, we can set for example $\ddot{\boldsymbol{q}} = [0, 25.0]^T$. Solving inverse dynamics we get $\boldsymbol{\lambda} = [79.4, 34.8]$. Note that a significant torque on joint 1 is needed in order to get joint 2 moving.

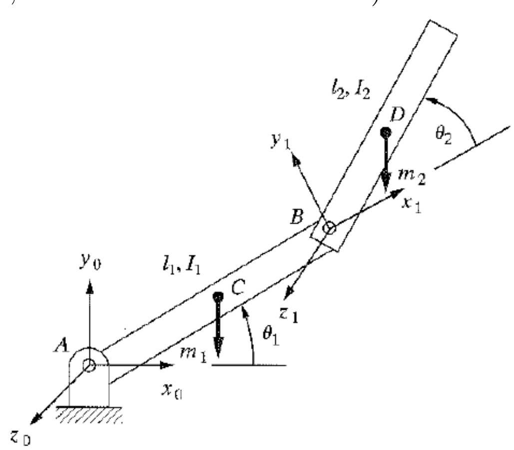
Pose 2. The pose in this example is $\boldsymbol{q} = [\pi/2, \pi/2]^T$.

At first, assume $\dot{\boldsymbol{q}} = [0, 0]^T$. For no desired acceleration ($\ddot{\boldsymbol{q}} = [0, 0]^T$), inverse dynamics gives us $\boldsymbol{\lambda} = [19.6, 0]$. This is natural, as joint 1 is holding up the arm against gravity.

Now assume that joint 2 is moving at some velocity, or $\dot{\boldsymbol{q}} = [0, 2]^T$. Again solving forward dynamics for joint torques, we get $\boldsymbol{\lambda} = [15.6, 0]$. Note that centripetal effects from joint 2 take some of the load off of joint 1.

1.4 Example: 2-link manipulator, distributed mass

We now consider a case where the mass of each link is no longer assumed to be concentrated at the far end, but instead is distributed equally along the link (a more realistic assumption). For each link, we thus assume that the center of mass is in the middle of the link (image and example from Niku, "Introduction to Robotics").



In this case, for computing the kinetic energy, we must take into account both the translational and rotational velocity of each link, and express them as a function of our variables $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$. Recall that the kinetic energy of a 2D body can be expressed as:

$$K = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2 \tag{29}$$

where v is the magnitude of the velocity of the center of mass, ω is the rotational velocity, and I is the *moment of inertia* computed about the center of mass.

For both links, the rotational velocities are straightforward:

$$\begin{aligned} \omega_1 &= \dot{q}_1 && (30) \\ \omega_2 &= \dot{q}_1 + \dot{q}_2 && (31) \end{aligned}$$

What about the velocities of the center of mass? We can, as in the previous example, perform forward kinematics to compute x_i and y_i (the Cartesian coordinates of the respective points), differentiate w.r.t. t to obtain $\dot{x}_i$ and $\dot{y}_i$, and then compute $v_i^2 = \dot{x}_i^2 + \dot{y}_i^2$. That yields:

$$\begin{aligned} v_1^2 &= 0.25 l_1^2 \dot{q}_1^2 && (32) \\ v_2^2 &= (l_1^2 + 0.25 l_2^2 + l_1 l_2 C_2)\dot{q}_1^2 + 0.25 l_2^2 \dot{q}_2^2 + (0.5 l_2^2 + l_1 l_2 C_2)\dot{q}_1 \dot{q}_2 && (33) \end{aligned}$$

We can plug all these back into (29) and compute the total kinetic energy of the system.

As in the previous case, the total potential energy can be computed as:

$$P = g m_1 \frac{l_1}{2} S_1 + g m_2 (l_1 S_1 + \frac{l_2}{2} S_{12}) \tag{34}$$

We can then compute the Lagrangian $L = K - P$ and differentiate it as in the previous case to obtain the relationships between $\boldsymbol{\lambda}$ and $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$.

In the previous examples, we've computed $\dot{x}_i$ and $\dot{y}_i$ as a function of $\dot{\boldsymbol{q}}$ for each link of the robot. In previous lectures, we've used the manipulator Jacobian to relate joint velocities to end-effector velocities. If we generalize that notion, we can define $\boldsymbol{J}_i$ as the matrix that relates joint velocities to the velocity of link i:

$$\boldsymbol{J}_i \dot{\boldsymbol{q}} = \dot{\boldsymbol{x}}_i \tag{35}$$

In this case, $v_i^2 = x_i^2 + y_i^2 = \dot{\boldsymbol{x}}_i^T \dot{\boldsymbol{x}}_i = \dot{\boldsymbol{q}}^T \boldsymbol{J}_i^T \boldsymbol{J}_i \dot{\boldsymbol{q}}$. We'll generalize this to three dimensions in the following sections.

1.5 Three-dimensional case

In three dimensions, the inertial moment of a body is replaced by the inertial tensor I, a 3x3 matrix. The inertial tensor generally depends on the reference frame it is computed in. A common practice is to compute the inertia tensor w.r.t. to a reference frame fixed to the body, or a body reference frame; then, it becomes strictly a property of the body shape and mass distribution.

However, in most cases, the velocity of a body is expressed in a different reference frame, such as the world frame. If R is the rotation matrix between the body reference frame and the world frame, then the inertia tensor can be transformed to the world frame using:

$$I_{\texttt{world}} = R I R^T \tag{36}$$

Therefore, if $\boldsymbol{v}$ is the translational velocity and $\boldsymbol{\omega}$ is the rotational velocity, both expressed in the world frame, the kinetic energy is:

$$K = \frac{1}{2} m \boldsymbol{v}^T \boldsymbol{v} + \frac{1}{2} \boldsymbol{\omega}^T I_{\texttt{world}} \boldsymbol{\omega} \tag{37}$$

In 3D, when also considering rotations, $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are both components of the 6x1 velocity vector $\dot{\boldsymbol{x}}$. Using the link Jacobian introduced earlier, we can compute these as:

$$\boldsymbol{J}_i \dot{\boldsymbol{q}} = \dot{\boldsymbol{x}} \tag{38}$$

or, broken down into translational and rotational components:

$$\begin{bmatrix} J_{vi} & J_{\omega i} \end{bmatrix} \dot{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} \tag{39}$$

Plugging into (37) we obtain:

$$K = \frac{1}{2} \dot{\boldsymbol{q}}^T \left[\sum_i m_i \boldsymbol{J}_{vi}(q) \boldsymbol{J}_{vi}^T(q) + \boldsymbol{J}_{\omega i}(q) R_i(q) I_i R_i^T(q) \boldsymbol{J}_{\omega i}^T(q) \right] \dot{\boldsymbol{q}} \tag{40}$$

$$= \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{q}}^T \tag{41}$$

We can go ahead to compute the Lagrangian

$$L = K - P = \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{q}}^T - \boldsymbol{P}(\boldsymbol{q}) \tag{42}$$

and use (4) to compute $\boldsymbol{\lambda}$. We won't go into the details of that computation here; however, we note that the result is of the general form:

$$\boldsymbol{\lambda} = \boldsymbol{M}(\boldsymbol{q}) \ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) \tag{43}$$

which we introduced earlier.

2 Newtonian Mechanics Approach

The Lagrangian approach we studied so far is based on an analysis of the energy of the system as a whole. It can be expressed directly in generalized variables (in our case, $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$). It is a very powerful and general way for finding the relationships between generalized variables and generalized forces acting on those variables. However, its formulation is... fairly involved.

There is an alternative formulation that can be easier to compute. Fundamentally, it simply says that for each link of the robot, Newton's second law must apply in both translation and rotation:

- The sum of **all the forces** applied to the link must be equal to the mass of the link times its acceleration. Note that the forces applied to the link include the force applied to the link by the previous link in the chain, as well as the force applied to it by the next link in the chain. These "inter-link forces" applied between links are originally unknown. There are other effects we can include: external forces, frictional forces, etc.

- The sum of **all the torques** applied to the link must be equal to the rotational inertia of the link times its angular acceleration. Note that the torques applied to the link include numerous terms. To begin with, we have the torques created by all the forces mentioned above. As before, there is a torque applied to the link by the previous link in the chain (which can include motor torque as a component), as well as the torque applied by the next link in the chain. These "inter-link torques" are originally unknown. In addition, we can include gyroscopic velocity-dependent effects.

This formulation adds many unknowns, mainly in the form of inter-link forces and torques. However, we also have more constraints: in addition to each link obeying Newton's second law (above), we also know that the robot stays connected, which means that the **forces, velocities and accelerations of consecutive links must be constrained to achieve this**. Once we also take into account these constraints, we have enough equations to balance all of our unknowns: if we know motor torques, we can compute joint accelerations (forward dynamics), or vice-versa (inverse dynamics).

2.1 Conventions

The exact equations we obtain from the Newtonian dynamics formulation is dependent on the coordinate frame being used as reference for each link. The general approach is, of course, valid in any coordinate frame, but the choice of frame might make the equations simpler or more complicated. We must also be very careful to be consistent in our choice of frames, and express all forces (and torques) using consistent assumptions.

Here is one possible convention; we will use it for the rest of the examples here. The coordinate frame for each link is at the joint preceding the link in the chain. We assume the center of mass is in the center of the link. We also choose to resolve all torques at the center of mass.

One link i, along with its proximal $(i-1)$ and distal $(i+1)$ links is shown below. We use the following conventions:

- The previous link $(i-1)$ is transmitting force $\boldsymbol{f}_i$ and torque $\boldsymbol{\mu}_i$ to link i through the joint between them. This implicitly means that the next link $(i+1)$ is transmitting force $-\boldsymbol{f}_{i+1}$ and torque $-\boldsymbol{\mu}_{i+1}$ to link i. Note that joint torque λ_i is only one component of $\boldsymbol{\mu}_i$ (the one along the joint axis).

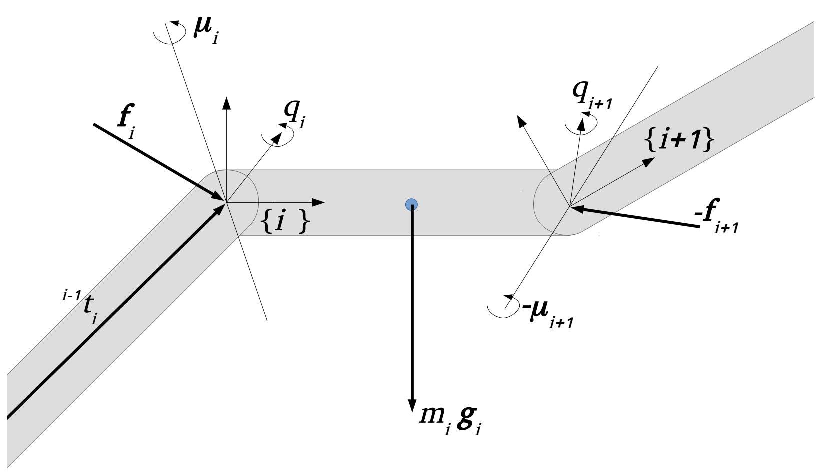
- ${}^{i}\boldsymbol{R}_{i+1}$ is the rotation matrix from the coordinate frame of link i to that of link $i+1$.

- ${}^{i-1}\boldsymbol{t}_i$ is the translation from the coordinate frame of link $i-1$ to that of link i, expressed in the {i-1} frame.

- ${}^{ci}\boldsymbol{t}_i$ is the translation from the center of mass of link i to the origin of the coordinate frame of link i, expressed in the {i} frame. Similarly, ${}^{ci}\boldsymbol{t}_{i+1}$ is the translation from the center of mass of link i to the origin of the coordinate frame of link i.

- $\boldsymbol{a}_{ci}$ is the acceleration of the center of mass of link i.

- $\boldsymbol{\omega}_i$ is the angular velocity of link i, and $\dot{\boldsymbol{\omega}}_i$ its angular acceleration. $\boldsymbol{I}_i$ is the inertial tensor of link i. Note that joint velocity $\dot{q}_i$ is only one component of $\boldsymbol{\omega}_i$ (the one along the joint axis), and a similar relationship exists between $\ddot{q}_i$ and $\dot{\boldsymbol{\omega}}_i$.



2.2 General Constraint Formulation

The force-based and torque-based instances of Newton's second law can be written as:

$$\begin{aligned}
\boldsymbol{f}_i - {}^iR_{i+1}\boldsymbol{f}_{i+1} + m_i\boldsymbol{g}_i &= m_i\boldsymbol{a}_{ci} && (44)\\
\boldsymbol{\mu}_i + ({}^{ci}t_i) \times \boldsymbol{f}_i - {}^iR_{i+1}\boldsymbol{\mu}_{i+1} - ({}^{ci}t_{i+1}) \times \left({}^iR_{i+1}\boldsymbol{f}_{i+1}\right) &= \boldsymbol{I}_i\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\boldsymbol{I}_i\boldsymbol{\omega}_i) && (45)
\end{aligned}$$

Note the presence of the gyroscopic term $\boldsymbol{\omega}_i \times (\boldsymbol{I}_i\boldsymbol{\omega}_i)$.

In addition, we can write the following relationships between the angular velocities and accelerations of consecutive links, taking into account the joint between them:

$$\begin{aligned}
\boldsymbol{\omega}_i &= {}^iR_{i-1}\boldsymbol{\omega}_{i-1} + \boldsymbol{z}_i\dot{q}_i && (46)\\
\dot{\boldsymbol{\omega}}_i &= {}^iR_{i-1}\dot{\boldsymbol{\omega}}_{i-1} + \boldsymbol{z}_i\ddot{q}_i + ({}^iR_{i-1}\boldsymbol{\omega}_{i-1} \times \boldsymbol{z}_i)\dot{q}_i && (47)
\end{aligned}$$

where $\boldsymbol{z}_i$ is the axis of rotation of joint i.

Finally, we can use these to compute the linear velocities of the link's reference frame, as well as its center of mass:

$$\begin{aligned}
\boldsymbol{a}_i &= {}^iR_{i-1}\left\{\boldsymbol{a}_{i-1} + \dot{\boldsymbol{\omega}}_{i-1} \times ({}^{i-1}t_i) + \boldsymbol{\omega}_{i-1} \times [\boldsymbol{\omega}_{i-1} \times ({}^{i-1}t_i)]\right\} && (48)\\
\boldsymbol{a}_{ci} &= \boldsymbol{a}_i + \dot{\boldsymbol{\omega}}_i \times ({}^it_{ci}) + \boldsymbol{\omega}_i \times [\boldsymbol{\omega}_i \times ({}^it_{ci})] && (49)
\end{aligned}$$

Note the presence of the centripetal terms of the type $\boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times t)$.

The unknowns here include:

- inter-link forces $\boldsymbol{f}_i$ and torques $\boldsymbol{\mu}_i$.
- link accelerations $\boldsymbol{a}_{ci}$ and $\dot{\boldsymbol{\omega}}_i$
- link angular velocities $\boldsymbol{\omega}_i$
- joint accelerations $\ddot{\boldsymbol{q}}_i$

We now have the following choice:

- If we prescribe known joint accelerations $\ddot{\boldsymbol{q}}_i$, the number of unknowns matches the number of constraints, and we can simply solve the resulting linear system for the all the other unknowns, including inter-link torques. We are thus performing inverse dynamics.
- Conversely, if we prescribe known motor torques, it means that one of the components of each $\boldsymbol{\mu}_i$ (the one aligning with the joint axis) is now known. We can thus solve the resulting linear system for all other unknowns, giving us the resulting joint accelerations. We are thus performing forward dynamics.

2.3 Recursive Algorithm

One downside of the formulation above is that it requires us to solve a linear system for both forward and inverse dynamics. Depending on the application, that might be considered an expensive operation. However, there are cases where we can get away without it.

We observe that the inverse dynamics case for an arm has an interesting property: the linear system is structured such that it can be solved with a simple recursive heuristic. The algorithm works as follows:

- **forward recursion**: for the first link ($i = 1$) of the robot, there is no previous link, hence $\boldsymbol{\omega}_{i-1} = \dot{\boldsymbol{\omega}}_{i-1} = 0$ and $\boldsymbol{a}_{i-1} = 0$. Hence, we can use (46), (47) and (48) to compute $\boldsymbol{\omega}_1$, $\dot{\boldsymbol{\omega}}_1$ and $\boldsymbol{a}_1$. Then, we can plug these into the same equations for link 2, and compute $\boldsymbol{\omega}_2$,$\dot{\boldsymbol{\omega}}_2$ and $\boldsymbol{a}_2$. We proceed like this until we've computed velocities and accelerations for all the links of the robot.

- **backwards recursion**: for the last link ($i = n$), the is no next link, hence $\boldsymbol{f}_{i+1} = 0$ and $\boldsymbol{\mu}_{i+1} = 0$. We can now use (44) and (45) to compute $\boldsymbol{f}_n$ and $\boldsymbol{\mu}_n$. Then, we plug these back into the same equations for link $n - 1$ and compute $\boldsymbol{f}_{n-1}$ and $\boldsymbol{\mu}_{n-1}$. We proceed like this until we've computed forces and moments for all the links of the robot.