

# Manipulator Equations for Systems with Constraints

Contact-aware Control, Lecture 4

by Sergei Savin

Fall 2020

- Lagrange equations
- Kinetic energy encoding (parts 1-4)
- Generalized inertia
  - ▶ Part 1-2
  - ▶ Part 3-4: Christoffel symbols
  - ▶ Part 5-6: Alternative formulation
- Generalized forces
  - ▶ Part 1, General case
  - ▶ Part 2, Conservative forces
  - ▶ Part 3-4, Reaction forces
- Manipulator equations
  - ▶ Part 1, no reaction forces
  - ▶ Part 2, reaction forces
- Read more
- Homework

Remember Lagrange equations:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial T}{\partial \mathbf{q}} = \tau \quad (1)$$

Let us remember that in the general case of point masses, kinetic energy is:

$$T = \sum 0.5 m_i \dot{\mathbf{r}}_i^\top \dot{\mathbf{r}}_i, \quad (2)$$

and in the general case of rigid bodies, it is:

$$T = \sum 0.5 m_i \dot{\mathbf{r}}_i^\top \dot{\mathbf{r}}_i + \sum 0.5 \mathbf{w}_i^\top \mathbf{I}_i \mathbf{w}_i, \quad (3)$$

Where  $\dot{\mathbf{r}}_i$  is the velocity of the center of mass of the  $i$ -th body, and  $\mathbf{w}$  is the angular velocity of that body.

# Kinetic energy encoding

## Part 1

Using chain rule we can describe the velocity of the center of mass:

$$\dot{\mathbf{r}}_i = \frac{\partial \mathbf{r}_i}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_i^v \dot{\mathbf{q}} \quad (4)$$

This establishes the connection between  $\dot{\mathbf{r}}_i$  and generalized velocities.

# Kinetic energy encoding

## Part 2

For the rotations, it is not as simple. We start by using a Poisson formula to connect rotation matrix  $\mathbf{T}(\mathbf{q})$  of a body to angular velocity of a body:

$$\mathbf{W}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T}\dot{\mathbf{T}}, \quad \dot{\mathbf{T}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{\partial \mathbf{T}}{\partial \mathbf{q}} \dot{\mathbf{q}} \quad (5)$$

where  $\mathbf{W}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & -\omega_1 & \omega_2 \\ \omega_1 & 0 & -\omega_3 \\ -\omega_2 & \omega_3 & 0 \end{bmatrix}.$

We can create notation:

$$\mathbf{w}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\mathbf{W}_{1,2} \\ \mathbf{W}_{1,3} \\ -\mathbf{W}_{2,3} \end{bmatrix} \quad (6)$$

# Kinetic energy encoding

## Part 3

### Homework 1

Prove that  $\mathbf{w}(\mathbf{q}, \dot{\mathbf{q}})$  is linear with respect to  $\dot{\mathbf{q}}$ .

Now we can *find angular velocity Jacobian* of (6) w.r.t.  $\dot{\mathbf{q}}$ :

$$\mathbf{J}_i^w = \frac{\partial \mathbf{w}_i}{\partial \dot{\mathbf{q}}} \quad (7)$$

Since  $\mathbf{w}_i$  is linear w.r.t.  $\dot{\mathbf{q}}$ , we can represent it as:

$$\mathbf{w}_i = \mathbf{J}_i^w \dot{\mathbf{q}} \quad (8)$$

# Kinetic energy encoding

## Part 4

Therefor me can rewrite the kinetic energy in terms of generalized velocity:

$$T = \sum 0.5 \dot{\mathbf{q}}^\top (\mathbf{J}_i^v)^\top m_i \mathbf{J}_i^v \dot{\mathbf{q}} + \sum 0.5 \dot{\mathbf{q}}^\top (\mathbf{J}_i^w)^\top \mathbf{I}_i \mathbf{J}_i^w \dot{\mathbf{q}} \quad (9)$$

Kinetic energy is a *quadratic form* of the generalized velocities.  
We can define the matrix of the quadratic form:

$$\mathbf{H} = \sum (\mathbf{J}_i^v)^\top m_i \mathbf{J}_i^v + \sum (\mathbf{J}_i^w)^\top \mathbf{I}_i \mathbf{J}_i^w \quad (10)$$

And therefor:  $T = 0.5 \dot{\mathbf{q}}^\top \mathbf{H} \dot{\mathbf{q}}$

# Generalized inertia

## Part 1

We can find derivatives of the kinetic energy (remembering that  $T = T^\top$ , and therefore  $\mathbf{H} = \mathbf{H}^\top$ ):

$$\frac{\partial T}{\partial \dot{\mathbf{q}}} = \mathbf{H} \dot{\mathbf{q}} \quad (11)$$

$$\frac{\partial T}{\partial \mathbf{q}} = 0.5 \dot{\mathbf{q}}^\top \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \dot{\mathbf{q}} \quad (12)$$

Notice that it is very tempting to say that  $0.5 \dot{\mathbf{q}}^\top \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \dot{\mathbf{q}} = 0.5 \dot{\mathbf{q}}^\top \dot{\mathbf{H}}$  but it is *not* the case.  $\frac{\partial \mathbf{H}}{\partial \mathbf{q}}$  is a three dimensional tensor, symmetric along the first and second dimension (so, transposing along these two dimensions doesn't change the products of the tensor with matrices or vectors). Multiplication by  $\dot{\mathbf{q}}$  happens along the first and second dimensions, while the partial differentiation happens along the third dimension, therefore the result is not necessarily equals to  $0.5 \dot{\mathbf{q}}^\top \dot{\mathbf{H}}$ .



Left-hand side of the Lagrange equations can be re-written as:

$$\frac{d}{dt} \left( \mathbf{H} \dot{\mathbf{q}} \right) - 0.5 \dot{\mathbf{q}}^\top \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \tau \quad (13)$$

We can expand the derivative of a product:

$$\mathbf{H} \ddot{\mathbf{q}} + \dot{\mathbf{H}} \dot{\mathbf{q}} - 0.5 \dot{\mathbf{q}}^\top \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \dot{\mathbf{q}} = \tau \quad (14)$$

Expression  $\dot{\mathbf{H}} \dot{\mathbf{q}} - 0.5 \dot{\mathbf{q}}^\top \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \dot{\mathbf{q}}$  is often cast as a linear form.  $\mathbf{C} \dot{\mathbf{q}}$   
The classic formula for calculating  $\mathbf{C} \dot{\mathbf{q}}$  uses Christoffel symbols.

Christoffel symbols-based formula for the  $\mathbf{C}\dot{\mathbf{q}}$  is:

$$\mathbf{C}\dot{\mathbf{q}} = \begin{bmatrix} \sum_{j,k}^n \Gamma_{1,j,k} \dot{q}_j \dot{q}_k \\ \dots \\ \sum_{j,k}^n \Gamma_{n,j,k} \dot{q}_j \dot{q}_k \end{bmatrix} \quad (15)$$

where Christoffel symbols  $\Gamma_{i,j,k}$  are given as:

$$\Gamma_{i,j,k} = \frac{1}{2} \left( \frac{\partial H_{i,j}}{\partial q_k} + \frac{\partial H_{i,k}}{\partial q_j} - \frac{\partial H_{k,j}}{\partial q_i} \right) \quad (16)$$

My apologies for not providing a derivation

# Generalized inertia

## Part 4, Christoffel symbols

Sometimes we need to find matrix  $\mathbf{C}$  specifically, rather than linear form  $\mathbf{C}\dot{\mathbf{q}}$ . This can be achieved using Christoffel symbols as well.

$$C_{i,j} = \sum_k^n \Gamma_{i,j,k} \dot{q}_k \quad (17)$$

# Generalized inertia

## Part 5, Alternative formulation

If you use auto-differentiation, you can consider directly using expression (14) to find  $\mathbf{C}\dot{\mathbf{q}}$ :

$$\mathbf{C}\dot{\mathbf{q}} = \dot{\mathbf{H}}\dot{\mathbf{q}} - 0.5 \frac{\partial \dot{\mathbf{q}}^\top \mathbf{H} \dot{\mathbf{q}}}{\partial \mathbf{q}} \quad (18)$$

In Matlab code it looks like:

```
0 C_times_v = reshape(dHdt*v, length(v), 1) - reshape  
    (jacobian((0.5*v' * H * v), q), length(v), 1);
```

# Generalized inertia

## Part 6, Alternative formulation

Alternatively, you can use the following formula:

$$\mathbf{C}\dot{\mathbf{q}} = \dot{\mathbf{H}}\dot{\mathbf{q}} - 0.5 \frac{\partial \text{vec}(\mathbf{H})}{\partial \mathbf{q}} (\dot{\mathbf{q}} \otimes \dot{\mathbf{q}}) \quad (19)$$

where  $\otimes$  is a Kronecker product, and  $\text{vec}()$  is vectorization of matrix (representing all its elements as a vector).

In Matlab code it looks like:

```
0 C_times_v = reshape(dHdt*v, [], 1) - 0.5*( jacobian(
    reshape(H, [], 1), q)'*kron(v, v) );
```

# Generalized forces

## Part 1, General case

You can express generalized forces of all kinds using discussed previously multiplication by the Jacobian:

$$\tau = \left( \frac{\partial \mathbf{r}_{\text{application point}}}{\partial \mathbf{q}} \right)^\top \mathbf{f}_{\text{ext}} \quad (20)$$

For a torque  $\xi$  applied to a rigid body, the corresponding generalized force is:

$$\tau = \mathbf{J}_w^\top \xi \quad (21)$$

where  $\mathbf{J}_w$  is the angular velocity Jacobian of that body. Note that both  $\mathbf{J}_w$  and  $\xi$  need to be expressed in the same basis.

# Generalized forces

## Part 2, Conservative forces

If the force is conservative, it is often easy to describe potential energy  $U$  associated with it. Then you can find the relevant generalized forces as:

$$\tau = -\frac{\partial U}{\partial \mathbf{q}} \quad (22)$$

Typically this is useful for gravitational forces and elastic forces.

# Generalized forces

## Part 3, Reaction forces

The discussion of the reaction forces stays the same as for any other general case forces, we use Jacobians to transform them into a generalized form.

We can define constraint Jacobian  $\mathbf{F}$  as:

$$\mathbf{F} = \frac{\partial \mathbf{g}(\mathbf{q})}{\partial \mathbf{q}} \quad (23)$$

Generalized reaction forces are found as:

$$\boldsymbol{\tau} = \mathbf{F}^\top \boldsymbol{\lambda} \quad (24)$$



# Generalized forces

## Part 4, Reaction forces

Sometimes a constraint can be expressed not as an explicit function of the Cartesian coordinates, but as an implicit one; for example, it can be expressed as a function of generalized coordinates.

### Example 1

$$\mathbf{g}(\mathbf{q}) = q_1 - q_3 + 2 = 0$$

### Example 2

$$\mathbf{g}(\mathbf{q}) = q_1^2 + q_2^2 - 1 = 0$$

But the generalized reaction forces for this case are still found the same way, as:

$$\boldsymbol{\tau} = \mathbf{F}^\top \boldsymbol{\lambda} \tag{25}$$

# Manipulator equations

## Part 1, no reaction forces

Finally we can write the form of manipulator equations:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} = \boldsymbol{\tau} \quad (26)$$

Another popular form specifically points out conservative forces  $\mathbf{p}$ :

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{p} = \boldsymbol{\tau}_{\text{non-conservative}} \quad (27)$$

The most concise and useful for this class form is:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \boldsymbol{\tau}_{\text{non-conservative}} \quad (28)$$

where  $\mathbf{c} = \mathbf{C}\dot{\mathbf{q}} + \mathbf{p}$ .

# Manipulator equations

## Part 2, reaction forces

Manipulator equations with reaction forces have the form:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} = \boldsymbol{\tau} + \mathbf{F}^\top \boldsymbol{\lambda} \quad (29)$$

The most concise and useful for this class form is:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \tau_{\text{non-conservative}} + \mathbf{F}^\top \boldsymbol{\lambda} \quad (30)$$

As usual, this is incomplete without the mention of the constraint:

$$\begin{cases} \mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \tau_{\text{non-conservative}} + \mathbf{F}^\top \boldsymbol{\lambda} \\ \mathbf{g}(\mathbf{q}) = 0 \end{cases} \quad (31)$$

You can read more at:

- [Chapter 4. Robot Dynamics and Control](#) - part 3.2, an interesting derivation.
- [Robot Dynamics Lecture Notes. Robotic Systems Lab, ETH Zurich HS 2017:](#)
  - ▶ [2.5 Angular Velocity](#)
  - ▶ [Chapter 3. Dynamics](#)
  - ▶ [3.4.2 Kinetic Energy](#)
  - ▶ [3.4.3 Potential Energy](#)
  - ▶ [3.4.5 Additional Constraints](#) (note some notational differences)
  - ▶ [3.5.2 Deriving Generalized Equations of Motion](#)

# Homework

Compute manipulator equations for a three link mechanism with fixed end effector (see figure). It should have tree generalized coordinates, two constraints. Preferably, use symbolic computations or auto-differentiation.



Lecture slides are available via Moodle.

You can help improve these slides at:

[github.com/SergeiSa/Contact-Aware-Control-Slides-Fall-2020](https://github.com/SergeiSa/Contact-Aware-Control-Slides-Fall-2020)

Check Moodle for additional links, videos, textbook suggestions.