Error dynamics and control of systems with constraints

Contact-aware Control, Lecture 6

by Sergei Savin

Fall 2020

Content

Trajectory tracking; desired trajectory

Trajectory tracking is a control problem that says:

Trajectory tracking

Find such *control law* $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ that solution of the dynamical system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t)$ converges to the *desired trajectory* $\mathbf{x}^* = \mathbf{x}^*(t)$.

For mechanical systems specifically we can write it as:

Trajectory tracking for mechanical systems

Find such control law $\mathbf{u} = \mathbf{u}(\mathbf{q}, \dot{\mathbf{q}}, t)$ that solution of the dynamical system $\mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{T}\mathbf{u}$ converges to the desired trajectory $\mathbf{q}^* = \mathbf{q}^*(t)$.

Desired trajectory and constraints

Assume your system is subject to constraints $\mathbf{g}(\mathbf{q}) = 0$, and you have desired trajectory $\mathbf{q}^* = \mathbf{q}^*(t)$. Then, unless $\mathbf{g}(\mathbf{q}^*(t)) = 0$, the desired trajectory is not valid.

You can find first two time derivatives of the desired trajectory: $\dot{\mathbf{q}}^*(t)$ and $\ddot{\mathbf{q}}^*(t)$. Defining $\mathbf{F} = \frac{\partial \mathbf{g}}{\partial \mathbf{q}}$ we can write first and second derivative of the constraint as: $\dot{\mathbf{g}}(\mathbf{q}) = \mathbf{F}\dot{\mathbf{q}}$ and $\ddot{\mathbf{g}}(\mathbf{q}) = \mathbf{F}\ddot{\mathbf{q}} + \dot{\mathbf{F}}\dot{\mathbf{q}}$. Therefor we can write implied conditions on the desired trajectory:

$$\dot{\mathbf{g}}(\mathbf{q}^*) = \mathbf{F}\dot{\mathbf{q}}^* = 0 \tag{1}$$

$$\ddot{\mathbf{g}}(\mathbf{q}^*) = \mathbf{F}\ddot{\mathbf{q}}^* + \dot{\mathbf{F}}\dot{\mathbf{q}}^* = 0 \tag{2}$$

We can rewrite equations of dynamics in the normal form:

$$\ddot{\mathbf{q}} = \mathbf{H}^{-1}(\mathbf{T}\mathbf{u} - \mathbf{c}) \tag{3}$$

Let us define *control error* $\mathbf{e}$ as follows:

$$\mathbf{e} = \mathbf{q}^{*} - \mathbf{q} \tag{4}$$

Then we can find its second derivative as: $\ddot{\mathbf{e}} = \ddot{\mathbf{q}}^{*} - \ddot{\mathbf{q}}$:

$$\ddot{\mathbf{e}} = \ddot{\mathbf{q}}^{*} - \mathbf{H}^{-1}(\mathbf{T}\mathbf{u} - \mathbf{c}) \tag{5}$$

If error dynamics is *stable*, it means the error will approach zero as the time approaches infinity. Is a good thing.

We can decide that we want error dynamics to have this form:

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = 0 \tag{6}$$

where $\mathbf{K}_d$ and $\mathbf{K}_p$ are diagonal positive-definite matrices. Equation (6) is stable. So, if we achieve that our error dynamics takes this form, we make it stable.

Let us use (5) to re-write (6):

$$\ddot{\mathbf{q}}^* - \mathbf{H}^{-1}(\mathbf{T}\mathbf{u} - \mathbf{c}) + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = 0 \tag{7}$$

So, we have:

$$\ddot{\mathbf{q}}^* - \mathbf{H}^{-1}(\mathbf{T}\mathbf{u} - \mathbf{c}) + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0 \tag{8}$$

Now we can multiply it by $\mathbf{H}$ (because it is invertible, so it's null space is trivial and we do not annihilate any part of the equation):

$$\mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) - (\mathbf{T}\mathbf{u} - \mathbf{c}) = 0 \tag{9}$$

...and then express $\mathbf{u}$ out:

$$\mathbf{u} = \mathbf{T}^+(\mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{c}) \tag{10}$$

This is called a *computed torque controller* (CTC), and it assumes that $(\mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{c})$ is in the column space of $\mathbf{T}$.

Thus CTC has the form: $\mathbf{u} = \mathbf{T}^{+}(\mathbf{H}(\ddot{\mathbf{q}}^{*} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \mathbf{c})$

We can separate feedback part $\mathbf{u}_{FB}$ and feedforward part $\mathbf{u}_{FF}$:

$$\mathbf{u} = \mathbf{u}_{FB} + \mathbf{u}_{FF} \tag{11}$$

$$\mathbf{u}_{FB} = \mathbf{T}^{+}\mathbf{H}(\mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) \tag{12}$$

$$\mathbf{u}_{FF} = \mathbf{T}^{+}(\mathbf{H}\ddot{\mathbf{q}}^{*} + \mathbf{c}) \tag{13}$$

Notice that feedback part is just a PD (proportional-derivative) controller with varying gains, while the feedforward part is just $\mathbf{u}$ expressed out of the robot's dynamics $\mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{T}\mathbf{u}$; the latter - finding $\mathbf{u}$ directly from the dynamics - is called *inverse dynamics*.

Dynamical system with constraints can be written as:

$$\begin{cases} \mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{T}\mathbf{u} + \mathbf{F}^{\top}\lambda \\ \mathbf{F}\ddot{\mathbf{q}} + \dot{\mathbf{F}}\dot{\mathbf{q}} = 0 \end{cases} \tag{14}$$

How do we apply the ideas about stable error dynamics here?

One naive approach is to define a new variable $\mathbf{v} = \mathbf{T}\mathbf{u} + \mathbf{F}^{\top}\lambda$ and rewrite the first equation in the system (14) as:

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{v} \tag{15}$$

Here it seems we can apply CTC directly. But we need to study how (and when) it will work.

We are considering equation $\mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{v}$. CTC for this case will take form:

$$\mathbf{v} = \mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{c} \tag{16}$$

But remember that $\mathbf{v} = \mathbf{T}\mathbf{u} + \mathbf{F}^\top\lambda$. We can always try to do our best to find such $\mathbf{u}$ that it holds, but what about λ?

On the value of λ

We should always keep in mind that λ is uniquely determined for any given $\mathbf{u}$. So, while we can't assign λ arbitrarily, as long as we assigned $\mathbf{u}$, we did in fact determined what value λ will take.

We remember from the previous lecture, that if we define $\mathbf{M} = \begin{bmatrix} \mathbf{H} & -\mathbf{F}^\top \\ \mathbf{F} & \mathbf{0} \end{bmatrix}$, then assuming $\mathbf{L}$ is a left inverse of $\mathbf{M}$, then we can write expressions for both $\ddot{\mathbf{q}}$ and λ in terms of its components:

$$\begin{cases} \ddot{\mathbf{q}} = \mathbf{L}_{11}(\mathbf{T}\mathbf{u} - \mathbf{c}) - \mathbf{L}_{12}\dot{\mathbf{F}}\dot{\mathbf{q}} \\ \lambda = \mathbf{L}_{21}(\mathbf{T}\mathbf{u} - \mathbf{c}) - \mathbf{L}_{22}\dot{\mathbf{F}}\dot{\mathbf{q}} \end{cases} \tag{17}$$

We can try to use this naive way of finding relation between λ and the control input $\mathbf{u}$, or one of the many more sophisticated ones. The idea would be to substitute the expression into the equation $\mathbf{v} = \mathbf{T}\mathbf{u} + \mathbf{F}^\top\lambda$, giving, in this case:

$$\mathbf{v} = \mathbf{T}\mathbf{u} + \mathbf{F}^\top(\mathbf{L}_{21}(\mathbf{T}\mathbf{u} - \mathbf{c}) - \mathbf{L}_{22}\dot{\mathbf{F}}\dot{\mathbf{q}}) \tag{18}$$

Alternatively, we can try to simultaneously calculate generalized accelerations $\ddot{\mathbf{q}}$, control inputs $\mathbf{u}$ and reaction forces λ. This allows us to bring in the constraint equation $\mathbf{F}\ddot{\mathbf{q}} + \dot{\mathbf{F}}\dot{\mathbf{q}} = 0$:

$$\begin{cases} \mathbf{H}\ddot{\mathbf{q}} + \mathbf{c} = \mathbf{v} \\ \mathbf{T}\mathbf{u} + \mathbf{F}^\top \lambda = \mathbf{v} \\ \mathbf{F}\ddot{\mathbf{q}} + \dot{\mathbf{F}}\dot{\mathbf{q}} = 0 \end{cases} \tag{19}$$

where unknowns are $\ddot{\mathbf{q}}$, $\mathbf{u}$ and λ. This is solved as a simple linear system:

$$\begin{bmatrix} \ddot{\mathbf{q}} \\ \mathbf{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{H} & 0 & 0 \\ 0 & \mathbf{T} & \mathbf{F}^\top \\ \mathbf{F} & 0 & 0 \end{bmatrix}^+ \begin{bmatrix} \mathbf{v} - \mathbf{c} \\ \mathbf{v} \\ -\dot{\mathbf{F}}\dot{\mathbf{q}} \end{bmatrix} \tag{20}$$

Column space

All possible outputs of a linear operator $\mathbf{A}$ are called *column space* of $\mathbf{A}$.

Null space

Null space of $\mathbf{A}$ is the set of all vectors $\mathbf{x}$ that $\mathbf{A}$ maps to 0

Now we can find all solutions to the system of equations $\mathbf{Ax} = \mathbf{0}$ by using functions that generate an orthonormal *basis* in the null space of $\mathbf{A}$. In MATLAB it is function `null()`.

In MATLAB it can be constructed by calling function `orth()`. Both `orth()` and `null()` (as well as `rank()` and `pinv()`) simply call `svd()` and perform minimal computations on the resulting decomposition. You can check it by typing `open orth` in MATLAB command window.

We can project any vector onto a subspace using a *projector*.

Definition 1

For linear space $\mathcal{L} \subset \mathbb{R}^n$, an orthogonal projector $\mathbf{P}$ onto it has properties:

- $\forall \mathbf{x} \in \mathbb{R}^n$, $\mathbf{P}\mathbf{x} \in \mathcal{L}$
- $\forall \mathbf{x} \in \mathcal{L}$, $\mathbf{P}\mathbf{x} = \mathbf{x}$
- $\forall \mathbf{y} \in \mathcal{L}$, $\mathbf{y}^\top (\mathbf{I} - \mathbf{P})\mathbf{x} = 0$

Projector $\mathbf{P}_c$ onto the column space of an operator $\mathbf{A}$ can be found as:

$$\mathbf{P}_c = \mathbf{A}\mathbf{A}^+ \tag{21}$$

Last expression suggests a simple feasibility condition for the existence of the control input that will generate the desired $\mathbf{v}$, namely that the left-hand-side vector of the linear system should lie in the column space of the matrix of the linear system:

$$\left(\mathbf{I} - \begin{bmatrix} \mathbf{H} & 0 & 0 \\ 0 & \mathbf{T} & \mathbf{F}^\top \\ \mathbf{F} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{H} & 0 & 0 \\ 0 & \mathbf{T} & \mathbf{F}^\top \\ \mathbf{F} & 0 & 0 \end{bmatrix}^+ \right) \begin{bmatrix} \mathbf{v} - \mathbf{c} \\ \mathbf{v} \\ -\dot{\mathbf{F}}\dot{\mathbf{q}} \end{bmatrix} = 0 \qquad (22)$$

We can make a simple set of necessary conditions. Remember that $\mathbf{v} = \mathbf{T}\mathbf{u} + \mathbf{F}^\top \lambda$. Therefore, vector $\mathbf{v}$ should lie in the column space of the matrix $\begin{bmatrix} \mathbf{T} & \mathbf{F}^\top \end{bmatrix}$:

$$\left(\mathbf{I} - \begin{bmatrix} \mathbf{T} & \mathbf{F}^\top \end{bmatrix} \begin{bmatrix} \mathbf{T} & \mathbf{F}^\top \end{bmatrix}^+\right) \mathbf{v} = 0 \tag{23}$$

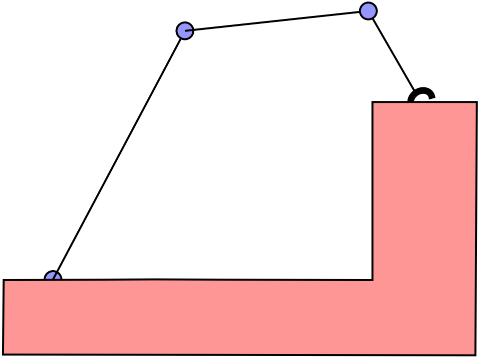
Read more

You can read more about Lagrange equation derivation at:

- *Slotine, J.J.E. and Li, W., 1987. On the adaptive control of robot manipulators. The international journal of robotics research, 6(3), pp.49-59* - learn more about error dynamics and similar techniques, is very interesting!

Homework

Write a tracking controller for this robot (description of the robot is given in the previous lectures).



Lecture slides are available via Moodle.

You can help improve these slides at:
github.com/SergeiSa/Contact-Aware-Control-Slides-Fall-2020

Check Moodle for additional links, videos, textbook suggestions.