MPC, Whole-body impulse control

Contact-aware Control, Lecture 11

by Sergei Savin

Fall 2023

Content

- Single rigid body model
- Bilinearity
- Linearizing rotation
- Model-predictive control
- Whole-body impulse control
- Quadruped control - big picture

Single rigid body model, 1

One of possible simplified models of walking robots is a single rigid body model. There are two ways to think about it:

- A robot consists of a body and legs. We assume that the mass and the inertia of the legs is much less than that of the body. This allows us to ignore the legs of the robot and model it as a body with reaction forces acting on it.
- Instead of modeling the robot using Lagrangian mechanics, we use a simplified model - we find a single rigid body model that describes the actual robot in the best possible way.

Single rigid body model, 2

Single rigid body model can be described as:

$$m\ddot{\mathbf{p}} = \sum \mathbf{f}_i + \mathbf{g} \tag{1}$$

$$\frac{d}{dt}(\mathbf{I}\omega) = \sum(\mathbf{r}_i \times \mathbf{f}_i) \tag{2}$$

where $\mathbf{p}$ is the position of the center of mass of the body, m and $\mathbf{I}$ is its mass and inertia, ω is its angular velocity, $\mathbf{f}_i$ are reaction forces, $\mathbf{r}_i$ are contact position and $\mathbf{g}$ is gravitational force.

Equation $m\ddot{\mathbf{p}} = \sum \mathbf{f}_i + \mathbf{g}$ shows a linear relation between acceleration $\ddot{\mathbf{p}}$ and reaction forces $\mathbf{f}_i$. Linear equations of this type can be used as a constraint in convex optimization problems (COP). COP usually allow linear equality constraints, but not other types of equality constraints.

Note that constraint $\frac{d}{dt}(\mathbf{I}\omega) = \sum(\mathbf{r}_i \times \mathbf{f}_i)$ is bilinear in position variables $\mathbf{r}_i$ and and reaction forces $\mathbf{f}_i$. This reflects the fact that torques depend linearly both on the force and point of application of this force. To use such constraint in COP, one of the two components has to be assumed to be constant - either point of application or the reaction force.

Bilinearity introduced by torques has a few important consequences:

- For trajectory planning problem (TPP) in full coordinates (or even when using SRB model) this means that the problem is not convex; TPP will be solved with non-linear solvers.
- For problems that need to be solved with convex solvers (such as MPC problems) the positions of contact points $\mathbf{r}_i$ is fixed, making r.h.s. of the equation $\frac{d}{dt}(\mathbf{I}\omega) = \sum(\mathbf{r}_i \times \mathbf{f}_i)$ linear in decision variables.

To linearize SRB dynamics we need to further introduce new variables and simplifications. Let $\Theta \in \mathbb{R}^3$ be encoding of robot orientation. Then we can simplify dynamics as:

$$\dot{\Theta} = \mathbf{R}\omega \tag{3}$$

$$\mathbf{I}_{\mathcal{G}}\dot{\omega} = \sum([\mathbf{r}_i]_\times \mathbf{f}_i) \tag{4}$$

Where $\mathbf{R}$ is orientation matrix, $\mathbf{I}_{\mathcal{G}} = \mathbf{R}\mathbf{I}_{\mathcal{B}}\mathbf{R}^\top$ is inertial tensor in the global frame, and $\mathbf{I}_{\mathcal{B}}$ - inertial tensor in the local frame.

Assuming discretization time step Δt, we get the following simplified discrete dynamics:

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \hat{\mathbf{f}}_k + \hat{\mathbf{g}} \tag{5}$$

where:
$\mathbf{x} = \begin{bmatrix} \Theta^\top & \mathbf{p}^\top & \dot{\Theta}^\top & \dot{\mathbf{p}}^\top \end{bmatrix}^\top$, $\hat{\mathbf{f}} = \begin{bmatrix} \mathbf{f}_1^\top & ... & \mathbf{f}_n^\top \end{bmatrix}^\top$,
$\hat{\mathbf{g}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{g}^\top \end{bmatrix}^\top$,

$$\mathbf{A} = \begin{bmatrix} \mathbf{1}_3 & \mathbf{0}_3 & \mathbf{R}\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{1}_3 & \mathbf{0}_3 & \mathbf{1}_3 \Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_3 & ... & \mathbf{0}_3 \\ \mathbf{0}_3 & ... & \mathbf{0}_3 \\ \mathbf{I}_{\mathcal{G}}^{-1} [\mathbf{r}_1]_\times \Delta t & ... & \mathbf{I}_{\mathcal{G}}^{-1} [\mathbf{r}_n]_\times \Delta t \\ \mathbf{1}_3 \Delta t / m & ... & \mathbf{1}_3 \Delta t / m \end{bmatrix}$$

Parameterizing reaction forces as $\mathbf{f}_j = \begin{bmatrix} f_{j,x} & f_{j,y} & f_{j,z} \end{bmatrix}^\top$ we can use a simple linear approximation of a friction cone:

$$|f_{j,x}| \leq \mu f_{j,z} \tag{6}$$
$$|f_{j,y}| \leq \mu f_{j,z} \tag{7}$$

To simplify notation we can designate friction cone constraint as $\hat{\mathbf{f}} \in \mathcal{C}$.

Finally, we formulate MPC as a quadratic program:

$$\begin{aligned} \underset{\mathbf{x}_{k+1}, \hat{\mathbf{f}}_k}{\text{minimize}} \quad & \sum ||\mathbf{x}_{k+1} - \mathbf{x}^{\text{ref}}_{k+1}||_{\mathbf{Q}} + \sum ||\hat{\mathbf{f}}_k||_{\mathbf{R}}, \\ \text{subject to:} \quad & \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \hat{\mathbf{f}}_k + \hat{\mathbf{g}}, \\ & \hat{\mathbf{f}} \in \mathcal{C}. \end{aligned}$$

where $||\mathbf{x}||_{\mathbf{Q}} = \sqrt{\mathbf{x}^\top \mathbf{Q} \mathbf{x}}$ - weighted norm, $\mathbf{Q}$, $\mathbf{R}$ - weight matrices and $\mathbf{x}^{\text{ref}}_{k+1}$ - reference trajectory.

This MPC can be run at 40 Hz. This gives us desired reaction forces, which we can use to compute joint torques.

We can write manipulator dynamics in the following way:

$$\mathbf{H}\begin{bmatrix}\ddot{\mathbf{q}}_F \\ \ddot{\mathbf{q}}_J\end{bmatrix} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} = \begin{bmatrix}\mathbf{0}_6 \\ \tau\end{bmatrix} + \mathbf{J}_c^\top \lambda \tag{8}$$

where $\mathbf{q}_F$ - floating-base coordinates and $\mathbf{q}_J$ - joint-space coordinates, $\mathbf{H}$ - generalized inertia matrix, $\mathbf{C}$ - Coriolis inertial force matrix, $\mathbf{g}$ - generalized gravitational force, τ - joint torques, $\mathbf{J}_c$ - constraint Jacobian and λ - reaction forces.

In order to compute joint torques, we can formulate the following quadratic problem:

$$
\begin{aligned}
\underset{\delta_a, \delta_f}{\text{minimize}} \quad & \delta_a^\top \mathbf{Q} \delta_a + \delta_f^\top \mathbf{R} \delta_f, \\
\text{subject to:} \quad & \mathbf{S}(\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g}) = \mathbf{S}\mathbf{J}_c^\top (\mathbf{f}_{\text{MPC}} + \delta_f), \\
& \ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{\text{des}} + \begin{bmatrix} \delta_a \\ \mathbf{0} \end{bmatrix} \\
& \mathbf{f}_{\text{MPC}} + \delta_f \in \mathcal{C}.
\end{aligned}
$$

where $\mathbf{S} = \begin{bmatrix} \mathbf{1}_{6\times 6} & 0 \end{bmatrix}$ is a floating-base selection matrix, $\mathbf{f}_{\text{MPC}}$ is a reference for the reaction force found from solving MPC and $\ddot{\mathbf{q}}_{\text{des}}$ is the desired acceleration found from inverse kinematics.

This controller can be solved at a rate of 500 Hz. Result of solving this QP is the value for decision variables δ_a, δ_f which allow us to compute the acceleration and reaction forces:

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}_{\text{des}} + \begin{bmatrix} \delta_a \\ \mathbf{0} \end{bmatrix} \tag{9}$$

$$\lambda = \mathbf{f}_{\text{MPC}} + \delta_f \tag{10}$$

Knowing both we can compute the torques using manipulator equations:

$$\begin{bmatrix} \mathbf{0}_6 \\ \tau \end{bmatrix} = \mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} - \mathbf{J}_c^\top \lambda \tag{11}$$

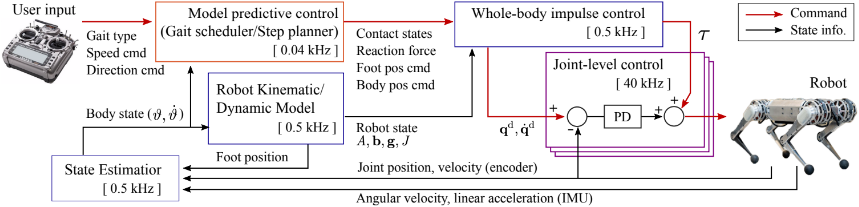

Figure 1: Control scheme proposed by Kim et al.

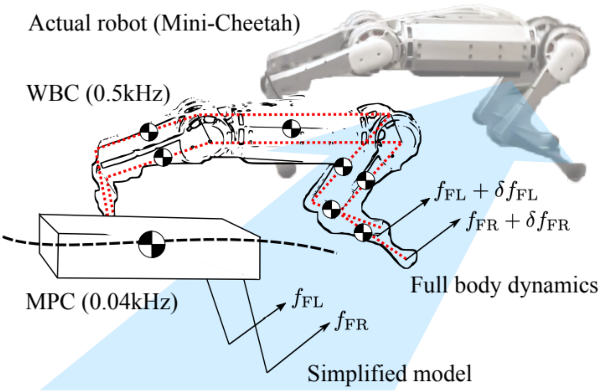

Figure 2: Sub-levels of the control system, as proposed by Kim et al.

Read more

- Kim, D., Di Carlo, J., Katz, B., Bledt, G. and Kim, S., 2019. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. arXiv:1909.06586

Lecture slides are available via Github, links are on Moodle

You can help improve these slides at:
github.com/SergeiSa/Contact-Aware-Control-Fall-2023