Trajectory Optimization with Optimization-Based Dynamics

Taylor Howell[1], Simon Le Cleac'h[1], Sumeet Singh[3], Pete Florence[3], Zachary Manchester[2], Vikas Sindhwani[3]
[1]Stanford University, [2]Carnegie Mellon University, [3]Google

abstract

We present a framework for **bi-level trajectory optimization** in which a system's **dynamics are encoded as the solution to a constrained optimization problem** and smooth gradients of this lower-level problem are passed to an upper-level trajectory optimizer. This optimization-based dynamics representation enables constraint handling, additional variables, and non-smooth behavior to be abstracted away from the upper-level optimizer, and allows classical unconstrained optimizers to synthesize trajectories for more complex systems. We provide a path-following method for efficient evaluation of constrained dynamics and **utilize the implicit-function theorem to compute smooth gradients** of this representation. We demonstrate the framework by modeling systems from locomotion, aerospace, and manipulation domains including: acrobot with joint limits, cart-pole subject to Coulomb friction, Raibert hopper, rocket landing with thrust limits, and planar-push task with optimization-based dynamics and then optimize trajectories using iterative LQR.

https://github.com/thowell/optimization_dynamics
https://arxiv.org/abs/2109.04928

key ideas

dynamics as constrained optimization problem

$$\underbrace{x_{t+1} \in z^*(\theta) = \underset{z \in \mathcal{K} \mid c(z;\theta)=0}{\arg\min} \ \ell(z;\theta)}_{= f_t(x_t, u_t)}$$

derivatives via implicit-function theorem

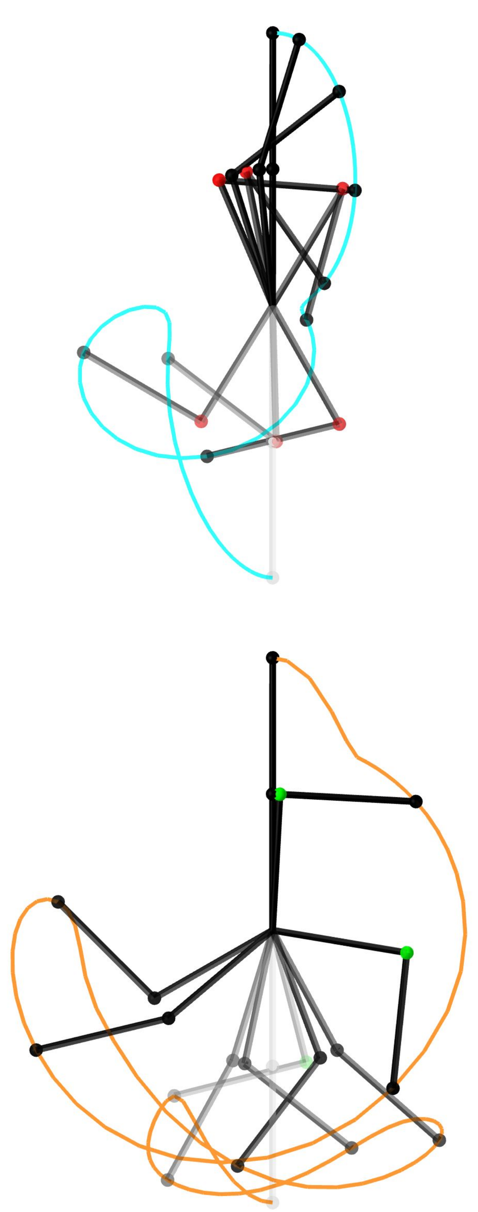
$$r(z;\theta) = \begin{cases} \partial \ell(z;\theta)/\partial z + (\partial c(z;\theta)/\partial z)^T \lambda - \nu = 0, \\ c(z;\theta) = 0, \\ z \circ \nu = \mu \mathbf{e}, \\ z \in \mathcal{K}, \ \nu \in \mathcal{K}^*, \end{cases} \longrightarrow \frac{\partial z}{\partial \theta} = -\left(\frac{\partial r}{\partial z}\right)^{-1} \frac{\partial r}{\partial \theta}$$

algorithm

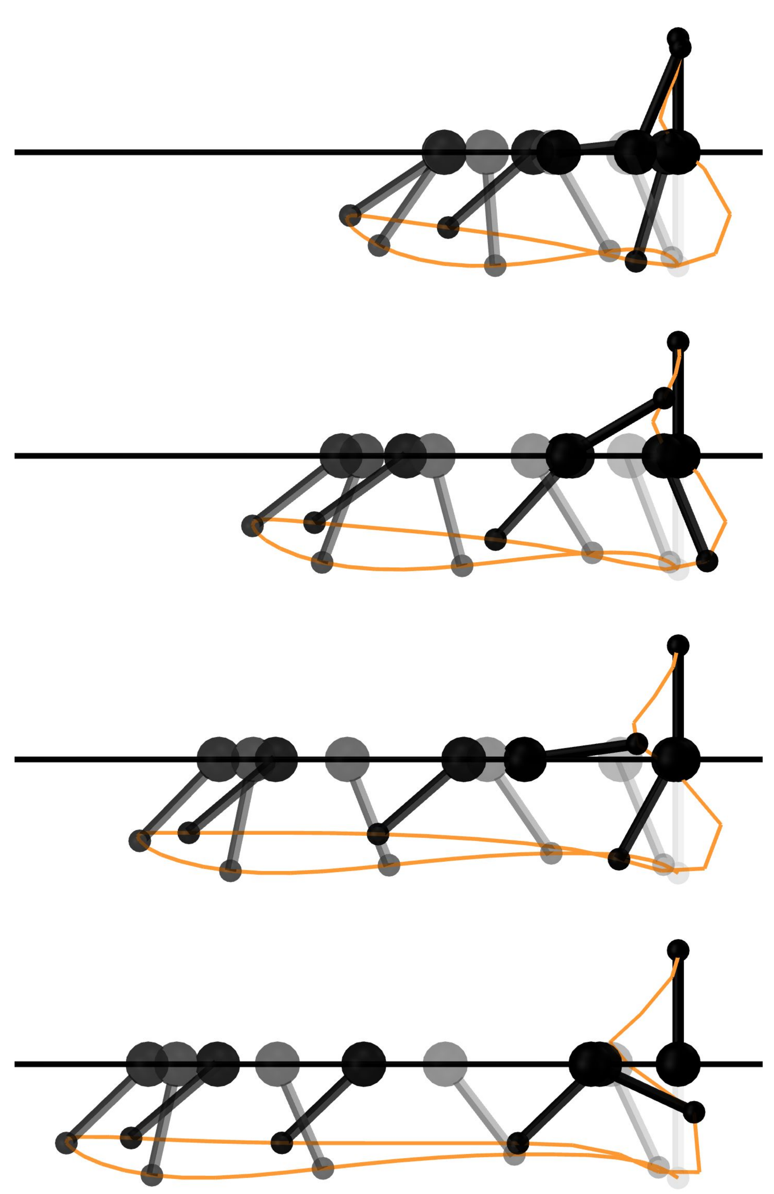
Algorithm 1 Differentiable Path-Following Method

procedure PATHFOLLOWING(z, θ)
  Parameters: $\beta = 0.5, \gamma = 0.1,$
    $\epsilon_\mu = 10^{-4}, \epsilon_r = 10^{-8}$
  Initialize: $\lambda = 0, \nu \in \mathcal{K}^*, \mu = 1.0, w_\mu = \{\}$
  $\bar{r} = r(w; \theta, \mu)$
  Until $\mu < \epsilon_\mu$ **do**
    $\Delta w = (\Delta z, \Delta \lambda, \Delta \nu) = (\partial r / \partial w)^{-1} \bar{r}$
    $\alpha \leftarrow 1$
    Until $z - \alpha \Delta z \in \mathcal{K}, \ \nu - \alpha \Delta \nu \in \mathcal{K}^*$ **do**
      $\alpha \leftarrow \beta \alpha$
    $\bar{r}_+ = r(w - \alpha \Delta w; \theta, \mu)$
    Until $\|\bar{r}_+\| < \|\bar{r}\|$ **do**
      $\alpha \leftarrow \beta \alpha$
      $\bar{r}_+ = r(w - \alpha \Delta w; \theta, \mu)$
    $w \leftarrow w - \alpha \Delta w$
    $\bar{r} \leftarrow \bar{r}_+$
    If $\|\bar{r}\| < \epsilon_r$ **do**
      $w_\mu \leftarrow w_\mu \cup w$
      $\mu \leftarrow \gamma \mu$
  $\partial w / \partial \theta \leftarrow \mathrm{IFT}(w_\mu, \theta)$
  Return $w, \partial w / \partial \theta$
end procedure

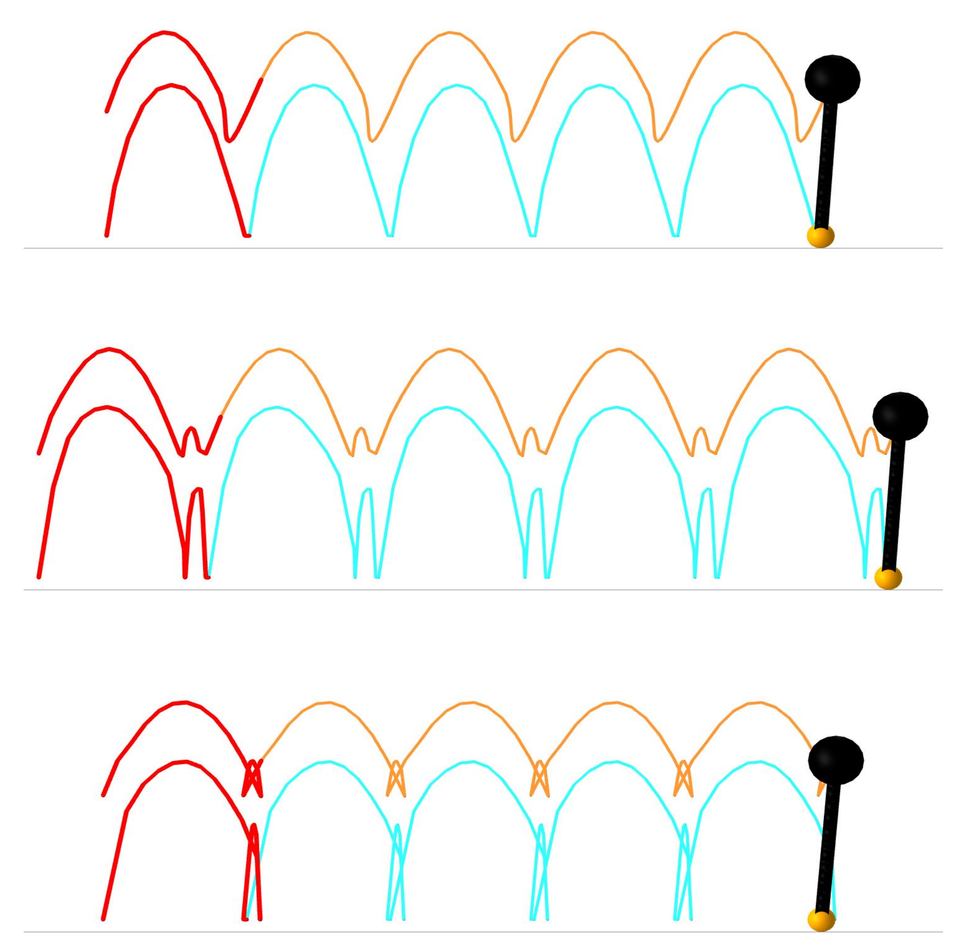
examples



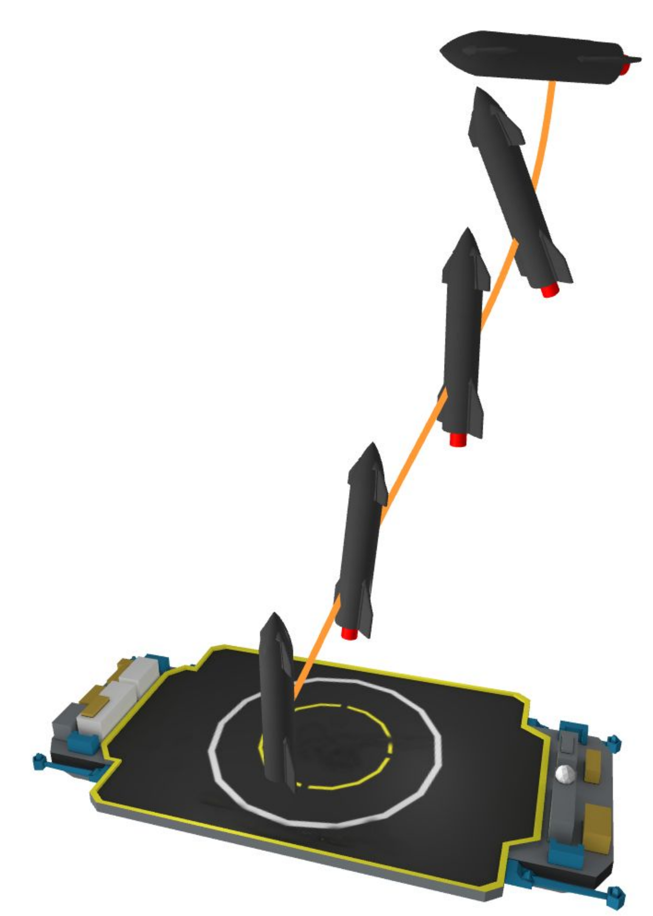
acrobot with joint limits



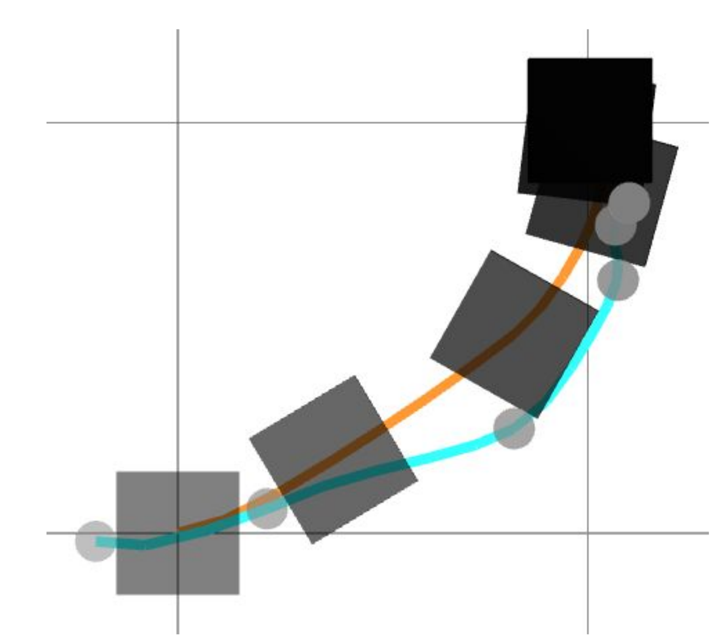
cart-pole experiencing Coulomb friction



hopper gait (contact-implicit)



rocket with thrust limits



planar push (contact-implicit)