Online Supplementary Materials For Rapid and Robust Trajectory Optimization for Humanoids

Bohao Zhang and Ram Vasudevan

Appendix I
Derivation of Theorem 4

The following proof is modified from [1, Section 8.8, 8.11, 8.12], with a more specific formulation of the projection matrix $G(q(t))$. Readers are recommended to refer to [1, Section 8] for a more comprehensive introduction to systems with closed-loop kinematic constraints.

For the sake of simplicity, we remove the time notation in this proof. We abuse the notation q and define it as a point in the robot configuration space $\mathcal{Q}$. The inertia matrix, the Coriolis matrix, and the gravity vector are then written as $H(q)$, $C(q,\dot{q})$, and $g(q)$, respectively. Hence, the dynamics equations are written as

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = Bu + J^T(q)\lambda. \tag{1}$$

Given Assumption 3, according to the inverse function theorem, for $\forall q_u \in \mathcal{Q}_u$ and $q_a \in \mathcal{Q}_a$, s.t. $c(q) = 0$, $J_u(q) \in \mathbb{R}^{n_u \times n_u}$ is invertible.

A. Derivation of (11) and (12)

The constraints are satisfied all the time, which means that the time derivative and the second-order time derivative are also zero. We first decompose (4) into unactuated part and actuated part:

$$\begin{aligned}\dot{c}(q) &= J(q)\dot{q} \\ &= J_u(q)\dot{q}_u + J_a(q)\dot{q}_a = 0.\end{aligned} \tag{2}$$

Solving for $\dot{q}_u$, we get

$$\dot{q}_u = -J_u^{-1}(q)J_a(q)\dot{q}_a. \tag{3}$$

The transform matrix $G(q) \in \mathbb{R}^{n \times n_a}$ is defined as

$$G_u(q) = -J_u^{-1}(q)J_a(q),\ G_a(q) = \mathbb{1}_{n_a \times n_a}, \tag{4}$$

where $G_u(q)$ and $G_a(q)$ are the unactuated columns and the actuated columns of $G(q)$, respectively. Hence,

$$\dot{q}_u = G_u(q)\dot{q}_a,\ \dot{q}_a = G_a(q)\dot{q}_a. \tag{5}$$

As a result, we can get (12) by concatenating two components above:

$$\dot{q} = G(q)\dot{q}_a. \tag{6}$$

By differentiating (6) with time, we can get (12).

B. Derivation of (13) and (14)

We first denote the uncontrained dynamics vector $H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q)$ using $\tilde{\tau}$. (1) then can be simplified to

$$\tilde{\tau} = Bu + J^T(q)\lambda \tag{7}$$

Simply consider the unactuated rows of (1):

$$\tilde{\tau}_u = B_u u + J_u^T(q)\lambda. \tag{8}$$

Since $B_u = \mathbb{0}_{n_u \times n_a}$, we can directly derive (13)

$$\tilde{\tau}_u = J_u^T(q)\lambda \tag{9}$$
$$\lambda = (J_u^{-1}(q))^T\tilde{\tau}_u \tag{10}$$

To get (14), first multiply $G^T(q)$ to both sides of (1):

$$G^T(q)\tilde{\tau} = G^T(q)Bu + G^T(q)J^T(q)\lambda. \tag{11}$$

For the right side of (11), since

$$\begin{aligned}B_u &= \mathbb{0}_{n_u \times n_a} \\ B_a &= \mathbb{1}_{n_a \times n_a},\end{aligned}$$

$G^T(q)Bu$ can be simplified:

$$\begin{aligned}G^T(q)Bu &= (G_a^T(q)B_a + G_u^T(q)B_u)u && \text{(12a)} \\ &= G_a^T(q)B_a u && \text{(12b)} \\ &= \mathbb{1}_{n_a \times n_a}u && \text{(12c)} \\ &= u. && \text{(12d)}\end{aligned}$$

$G^T(q)J^T(q)\lambda$ can be simplified:

$$\begin{aligned}G^T(q)J^T(q)\lambda &= (J(q)G(q))^T\lambda && \text{(13a)} \\ &= (J_a(q)G_a(q) + J_u(q)G_u(q))^T\lambda && \text{(13b)} \\ &= (J_a(q) - J_u(q)J_u^{-1}(q)J_a(q))^T\lambda && \text{(13c)} \\ &= (J_a(q) - J_a(q))^T\lambda && \text{(13d)} \\ &= \mathbb{0}_{n_a \times n_u}\lambda && \text{(13e)} \\ &= \mathbb{0}_{n_a}. && \text{(13f)}\end{aligned}$$

Hence (11) is finally simplified to

$$u = G^T(q)\tilde{\tau} \tag{14}$$
$$= G^T(q)(H(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q)), \tag{15}$$

which gives us the same equation in (14). □

Appendix II
Ablation Studies on Hyperparameters of RAPTOR

This section serves as an ablation study on RAPTOR parameters (degree of the Bezier curve V and the number of time nodes N) and Ipopt parameters (`mu_strategy` and `linear_solver`). The maximum iteration for Ipopt is set to 100 for this experiment. The results are listed in Table I. We provide 100 random initial guesses by sampling in the range of $[-0.1, 0.1]$ for all entries of the decision variable vector. We are able to show that RAPTOR is robust toward random initial guesses, that it converges to a feasible solution in most cases.

We can conclude that the "monotone" strategy generally yields more robust convergence than the "adaptive" strategy, though not converging well in some cases. The linear solver ma57 is faster than ma27 in smaller problems ($V \leq 7$), but does not show much difference in larger problems ($V = 8$). The best performance in terms of both cost and constraint violation is with ma86, but it also takes much longer in terms of computation time.

Appendix III
Comparison Results on Talos

We also performed the same comparison on another humanoid Talos that was originally included in the official examples of aligator and does not contain any closed-loop kinematics chains.

References

[1] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

V and N	"mu_strategy"	"linear_solver"	Cost	Constraint Violation	Computation Time (sec)
$V=5,\ N=14$	adaptive	ma27	(4854, 1.002e+04)	(0.02192, 158.3)	(8.614, 18.55)
		ma57	(4822, 1.175e+04)	(0.01986, 69.54)	(4.762, 37.49)
		ma86	(1373, 1.6e+04)	(5.362e-06, 1.974e+04)	(7.505, 21.39)
	monotone	ma27	(1702, 1.041e+04)	(5.535e-06, 1.263e+04)	(8.175, 18.13)
		ma57	(1702, 1.041e+04)	(5.466e-06, 1.263e+04)	(2.765, 35.58)
		ma86	(1734, 1.064e+04)	(7.231e-06, 1.269e+04)	(7.765, 19.9)
$V=6,\ N=18$	adaptive	ma27	(5874, 2.544e+04)	(0.01342, 170.2)	(14.36, 16.09)
		ma57	(6185, 2.543e+04)	(0.01342, 170.2)	(13.56, 33.82)
		ma86	(2231, 5.936e+04)	(5.829e-06, 9245)	(25.34, 52.21)
	monotone	ma27	(1932, 1.161e+04)	(3.853e-06, 1.223e+04)	(13.34, 18.44)
		ma57	(1932, 1.161e+04)	(4.191e-06, 1.223e+04)	(10.5, 24.24)
		ma86	(1906, 1.523e+04)	(3.881e-06, 1.224e+04)	(21.91, 52.96)
$V=7,\ N=22$	adaptive	ma27	(1.04e+04, 2.928e+04)	(0.02504, 194.8)	(23.02, 26.34)
		ma57	(1.054e+04, 4.186e+04)	(0.02488, 194.8)	(5.512, 46.93)
		ma86	(8489, 2.956e+04)	(0.01018, 165.7)	(89.26, 96.56)
	monotone	ma27	(1960, 1.629e+04)	(3.517e-06, 1.51e+04)	(21.05, 22.37)
		ma57	(1959, 1.629e+04)	(3.517e-06, 1.51e+04)	(4.79, 23.82)
		ma86	(1951, 1.606e+04)	(4.183e-06, 2.102e+04)	(32.11, 93.27)
$V=8,\ N=26$	adaptive	ma27	(1.773e+04, 3.924e+04)	(0.02039, 157.7)	(34.24, 36.85)
		ma57	(1.599e+04, 3.924e+04)	(0.01586, 157.7)	(42.05, 114.9)
		ma86	(1.304e+04, 4.256e+04)	(0.008349, 2416)	(122.8, 127.1)
	monotone	ma27	(2745, 2.081e+04)	(6.336e-06, 2.747e+04)	(32.51, 42.64)
		ma57	(2745, 2.081e+04)	(6.336e-06, 2.747e+04)	(31.07, 66.25)
		ma86	(2651, 2.09e+04)	(5.563e-06, 2.751e+04)	(49.62, 125.8) $\pm$

TABLE I: Ablation studies on hyperparameters of RAPTOR based on Digit. We report (median, maximum) values of the cost, the constraint violation, and the computation time in seconds given 100 random initial guesses for all comparisons.

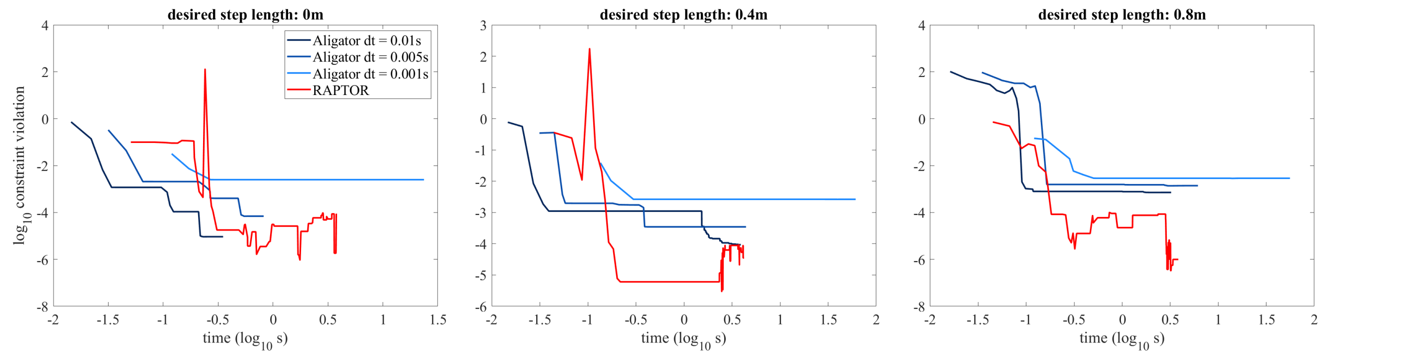

Fig. 1: Pareto curve of the constraint violation with respect to the computation time. Note that all curves do not start from 0 in time, since we need to evaluate the problem for at least one iteration to get the constraint violation. In other words, the beginning of the curves indicates the computation time of the first iteration.

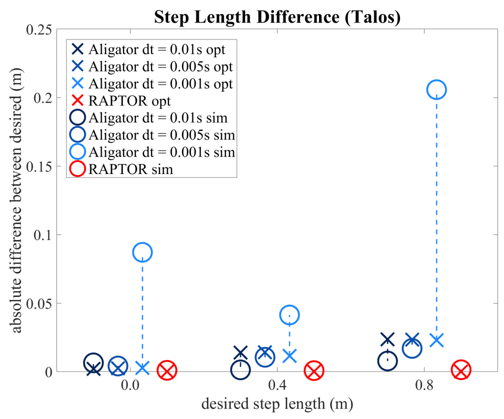

Fig. 2: The absolute value of the difference between the desired step length and the actual step length. The crosses indicate the difference on the trajectory generated by the optimization. They may be away from 0 due to optimization not converging to a feasible solution. The circles indicate the difference on the trajectory in simulation when tracking the optimized trajectory using a controller. The circles may not be aligned with the crosses due to imperfect tracking of the controller.

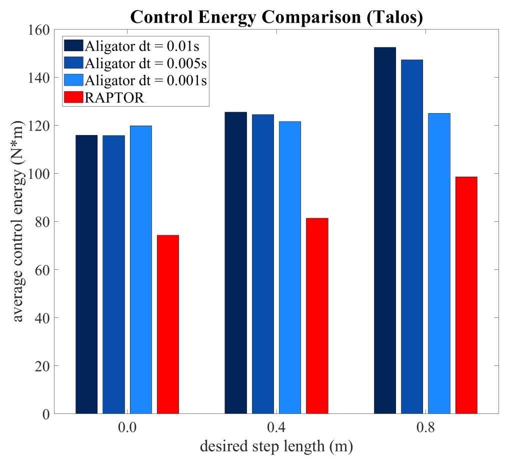

Fig. 3: Control energy consumption of 3 variants of aligator and RAPTOR in the simulation for 3 different desired step lengths