Discovery of Dynamic Loco-manipulation Behaviors

Michal Ciebielski[1,2], Haizhou Zhao[1,3], Aaron Johnson[2,4], Majid Khadiv[1]

***Abstract*— Dynamic loco-manipulation behavior discovery can be cast as a search over contact-mode sequences, whose transitions must satisfy the hybrid dynamics induced by the different contact modes. The key challenge is combinatorial growth in the number of candidate sequences with the sequence horizon. Prior approaches address this by planning with a predetermined set of higher level actions, however doing so narrows the set of behaviors that can be discovered. We instead preserve a broad search space and make it tractable by using an efficient kinematic exploration strategy. Specifically, we expand a tree of contact-mode sequences using a sequential kinematic feasibility check and invoke full trajectory optimization only for sequences that reach the goal. This defers expensive full-order trajectory optimization until a goal-reaching sequence is found, reducing the number of full-order solves while enabling the automatic discovery of diverse dynamic loco-manipulation behaviors.**

I. Introduction

Generation of loco-manipulation behaviors remains a challenging problem due to the combinatorial complexity of the sequential interaction problem and the difficulty of controlling the underactuated dynamics of both the robot and the manipulated objects. Despite this challenge, any method aiming to capture the full range of loco-manipulation behaviors must model both the full system dynamics and the space of possible sequential interactions among the robot, the manipulated object, and the environment.

Many current approaches to humanoid control avoid the complexity of algorithmic motion planning by exploiting the similarities between human and humanoid morphology to retarget human demonstrations [1], [2], [3], [4], [5]. This is a promising direction because it reduces much of the complexity previously needed to enable control of humanoid systems. When combined with reinforcement learning, the resulting policies can be executed robustly [1], [2], [3], [4]. Recent work has also focused on training on large corpora of demonstrations to learn policies that generalize across a broader range of motions [4].

While recent work largely relies on retargeting and imitation, we instead study the automatic generation of novel behaviors. We begin by formalizing a discrete search space over contact modes and then defining the constraints required for feasible transitions between them. This formulation results in a graph structure that can be traversed to address a variety of planning problems. Since the graph is combinatorially large, we use a sequential kinematic optimization to grow a search tree of kinematically feasible contact-mode transitions, and validate goal-reaching sequences using trajectory optimization. We demonstrate that this method can discover a diverse set of solutions (video link[5]), on two variants of an object-placement task. We also show that the proposed kinematic-feasibility pruning strategy produces a negligible number of false negatives, making it an effective proxy for pruning the search tree efficiently.

In section II we formulate the problem and then in section III we propose our approach to solve the resulting optimization. Next in section IV we present and discuss our results and we conclude our findings in section V.

II. Problem Formulation

A. Problem statement

We study dynamic loco-manipulation planning problems in which task completion requires a sequence of contact-mode transitions. While the combinatorial set of candidate contact modes can be enumerated a priori, the subset of modes and transitions that are actually feasible is not known. We therefore introduce a contact-mode graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$, where $\mathcal{C}$ denotes candidate contact modes and $\mathcal{E}$ candidate transitions between them. Since feasibility depends on the system dynamics, the feasible subgraph of $\mathcal{G}$ must be uncovered during the search rather than specified in advance.

The planning problem is thus a hybrid graph search rooted at an initial contact mode coupled with an initial continuous state. Subsection II-B defines the graph nodes, while Subsection II-C formalizes node transition feasibility. The objective is then to find a goal-reaching contact-mode sequence and associated continuous trajectories that satisfy these conditions and minimize the cost in Subsection II-D.

B. Contact Mode Definition

Let $\mathcal{I}$ be the set of all interfaces (end effector patches, object surfaces, environment planes, etc.) in the scene, and let $\mathcal{I}^{\text{act}} \subseteq \mathcal{I}$ denote the subset of active interfaces, namely those whose contact state may change. Let Θ be the set of admissible contact types (e.g. free, sticking, sliding, etc), and let $\varnothing$ denote the free state. For each active interface $i \in \mathcal{I}^{\text{act}}$, we define the contact state of interface i as

$$m_i = (i, j_i, \theta_i), \qquad j_i \in \mathcal{I} \cup \{\varnothing\}, \;\; \theta_i \in \Theta \cup \{\varnothing\}. \quad (1)$$

This work was funded in part by SIEMENS AG and the Technical University of Munich - Institute for Advanced Study and by the Huawei- TUM joint laboratory.

[1]Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich (TUM), Germany. firstname.lastname@tum.de

[2]Institute for Advanced Study, Technical University of Munich, Garching, Germany.

[3]Tandon School of Engineering, New York University, USA. hz3862@nyu.edu

[4]Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA. amj1@andrew.cmu.edu

[5]https://www.youtube.com/watch?v=PlkZBepd4j0

Here, j_i specifies the interaction partner of i, while θ_i specifies the contact type. The partner interface j_i provides the geometric context of the interaction but is not itself associated with a force decision variable in this representation. The special value $\varnothing$ denotes that interface i is free, in which case no interaction partner is defined. Accordingly, we require

$$\theta_i = \varnothing \iff j_i = \varnothing. \tag{2}$$

A contact mode is then the assignment of a contact state to every active interface, i.e.,

$$c = \{(i, j_i, \theta_i) \mid i \in \mathcal{I}^{\mathrm{act}}\}. \tag{3}$$

Equivalently, $c \in \mathcal{C}$ may be viewed as a map from active interfaces to partner–type assignments. The set $\mathcal{C}$ defines the node set of the candidate graph $\mathcal{G}$. The next subsection specifies the conditions under which transitions between these nodes are physically realizable.

C. Contact Mode Transition Feasibility

For a contact mode c, let $\mathcal{M}(c)$ denote the feasible constraint manifold associated with it,

$$\mathcal{M}(c) := \left\{ (x, \dot{x}, u) \;\middle|\; \begin{array}{l} \mathrm{dynamics}(x, \dot{x}, u, c) = 0, \\ \mathrm{contact}(x, u, c) \le 0, \\ \mathrm{collision}(x) \le 0, \\ \mathrm{limits}(x, u) \le 0 \end{array} \right\} \tag{4}$$

The set $\mathcal{M}(c)$ contains all state, state-derivative, and control tuples that satisfy the dynamics and contact conditions induced by c, remain within hardware limits, and avoid unintended collisions. The latter requirement is particularly important, since any additional collision would create an unmodeled contact and thus imply a departure from the contact mode c.

A transition from c to c' is feasible only if there exist a duration $\tau > 0$, continuous trajectories

$$\mathbf{x} : [0, \tau] \to \mathcal{X}, \qquad \mathbf{u} : [0, \tau] \to \mathcal{U},$$

and a contact-mode evolution $c(t)$ consistent with the transition from c to c', such that

$$\mathbf{x}(0) = x, \qquad \mathbf{x}(\tau) = x',$$

and

$$(\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t)) \in \mathcal{M}(c(t)) \qquad \forall t \in [0, \tau].$$

D. Optimization Formulation

We now combine the discrete contact-mode search with the continuous feasibility conditions into a hybrid optimal control problem. The decision variables are both the continuous robot-object trajectories and the discrete contact-mode sequence. Let $x : [0, T] \to \mathcal{X}$ and $u : [0, T] \to \mathcal{U}$ denote the state and control trajectories of a floating-base robot with n joints manipulating m objects, with

$$x := (x^r, x^{o_1}, \ldots, x^{o_m}), \qquad u := (u^r, u^{o_1}, \ldots, u^{o_m}). \tag{5}$$

Here, the robot state is $x^r := (q^r, v^r, h^r)$, where $q^r \in \mathbb{SE}(3) \times \mathbb{R}^n$, $v^r \in \mathbb{R}^{n+6}$, and $h^r \in \mathbb{R}^6$ denote the robot configuration, generalized velocity, and centroidal momentum, respectively. Each object state is $x^{o_i} := (q^{o_i}, \mathcal{V}^{o_i})$, where $q^{o_i} \in \mathbb{SE}(3)$ and $\mathcal{V}^{o_i} \in \mathbb{R}^6$ denote the object pose and body velocity. The robot control is $u^r := (\dot{v}^r, \lambda^r_{ee_1}, \ldots, \lambda^r_{ee_{n_{ee}}})$, and each object control is $u^{o_i} := (\dot{\mathcal{V}}^{o_i}, \mathcal{W}^{o_i}_{env})$.

Let $c_{0:K}$ denote the contact-mode sequence and $\bar{T}_{0:K-1}$ the corresponding stage durations. These durations induce switching times

$$t_0 := 0, \qquad t_{k+1} := t_k + \bar{T}_k, \tag{6}$$

so that the mode schedule is piecewise constant with $c(t) = c_k$ for $t \in [t_k, t_{k+1})$, and the final time is $T = t_K$. The successor map $\Gamma(c_k) \subseteq \mathcal{C}$ denotes the set of admissible modes that may follow c_k, thereby encoding the allowable discrete transitions in the search graph. With these definitions, the complete planning problem is

$$\begin{aligned} \min_{\substack{x(\cdot),\, u(\cdot),\\ c_{0:K},\, \bar{T}_{0:K-1}}} \quad & \int_0^T \phi(x(t), u(t))\, dt + \phi_T(x(T)) + w_T T \\ \text{s.t.} \quad & x(0) = x_{\mathrm{init}}, \quad c_0 = c_{\mathrm{init}}, \quad x(T) \in X_{\mathrm{goal}}, \\ & \forall k \in \{0, \ldots, K-1\} : \\ & \quad c_{k+1} \in \Gamma(c_k), \quad \bar{T}_{\min} \le \bar{T}_k \le \bar{T}_{\max}, \\ & \quad (x(t), \dot{x}(t), u(t)) \in \mathcal{M}(c_k), \quad \forall t \in [t_k, t_{k+1}]. \end{aligned} \tag{7}$$

where ϕ is the stage cost, ϕ_T is the terminal cost, ω_T is the time penalty. Equation (7) defines the continuous-time hybrid planning problem over contact-mode sequences and their associated state and control trajectories. Then next section III introduces the finite-dimensional transcription used in practice, together with the pruning strategy, trajectory optimization, and tree-search procedures used to explore the contact-mode graph efficiently.

III. Method

The hybrid planning problem in (7) is difficult to solve directly because it couples a combinatorial search over contact-mode sequences with highly nonlinear continuous dynamics. Our approach addresses this by combining discrete search over contact mode sequences with continuous optimization over the state and control trajectories. The next two subsections present the optimization problems associated with a fixed contact-mode sequence: first a sequential kinematic feasibility problem III-A, and then a full trajectory optimization III-B. We then show how these components are combined within a tree-search procedure III-C.

A. Sequential Kinematic Feasibility

Given a contact-mode sequence, the sequential kinematic feasibility problem solves for robot-object configurations q that satisfy the sequential contact, collision, and configuration-limit constraints associated with each mode (8). At this stage, no time-dependent dynamics are imposed. The optimization only tests whether the sequence is geometrically feasible. In addition to these per-mode constraints,

adjacent modes are coupled through transition-consistency constraints detailed below.

$$\begin{aligned}
\min_{q_{0:N-1}} \quad & \sum_{i=0}^{N-1} (q_i - q_{\text{nom}})^\top W (q_i - q_{\text{nom}}) \\
\text{s.t.} \quad & \forall i \in \{0, \dots, N-1\} : \\
& \quad \text{contact}(q_i, c_i) \le 0, \\
& \quad \text{collision}(q_i) \le 0, \\
& \quad \text{limits}(q_i) \le 0, \\
& \forall i \in \{0, \dots, N-2\} : \\
& \quad \forall k \in \mathcal{I}^{\text{act}} \text{ satisfying (9a) or (9b)} : \\
& \qquad \text{transitionConsistency}_{i,k}(q_i, q_{i+1}) = 0.
\end{aligned} \tag{8}$$

For each transition $i \to i+1$, the constraint $\text{transitionConsistency}_{i,k}(q_i, q_{i+1}) = 0$ is imposed only for active interfaces $k \in \mathcal{I}^{\text{act}}$ whose contact-state components in c_i and c_{i+1} satisfy one of the following conditions:

$$(j_k^i, \theta_k^i) = (j_k^{i+1}, \theta_k^{i+1}) \neq (\varnothing, \varnothing), \tag{9a}$$

$$(j_k^i, \theta_k^i) \neq (\varnothing, \varnothing), \qquad (j_k^{i+1}, \theta_k^{i+1}) = (\varnothing, \varnothing). \tag{9b}$$

The first case corresponds to maintaining the same contact across the transition, in which case the relative contact pose must be preserved. The second corresponds to releasing an existing contact. Here, (k, j_k^i, θ_k^i) and $(k, j_k^{i+1}, \theta_k^{i+1})$ denote the components of c_i and c_{i+1}, respectively.

B. Trajectory Optimization

While the sequential kinematic feasibility problem only checks geometric consistency for a fixed contact-mode sequence, the trajectory optimization enforces the full robot-object dynamics. For a given contact-mode sequence, it solves for dynamically feasible state and control trajectories, together with the duration of each contact stage. This optimization therefore serves as the full-order validation of a candidate sequence.

The continuous-time problem in (7) is transcribed into the following finite-dimensional nonlinear program:

$$\begin{aligned}
\min_{\substack{x_{0:N},\, u_{0:N-1},\\ \bar{T}_{0:K-1}}} \quad & \sum_{i=0}^{N-1} \phi(x_i, u_i) + \phi_N(x_N) + w_T \sum_{k=0}^{K-1} \bar{T}_k \\
\text{s.t.} \quad & x_0 = x_{\text{init}}, \quad x_N \in X_{\text{goal}}, \\
& \text{terminal}(x_N, c_K) \le 0, \\
& \forall k \in \{0, \dots, K-1\} : \\
& \quad \bar{T}_{\min} \le \bar{T}_k \le \bar{T}_{\max}, \\
& \forall i \in \{0, \dots, N-1\} : \\
& \quad \text{dynamics}(x_i, x_{i+1}, u_i, \bar{T}_{k(i)}, c_{k(i)}) = 0, \\
& \quad \text{contact}(x_i, u_i, c_{k(i)}) \le 0, \\
& \quad \text{collision}(x_i) \le 0, \\
& \quad \text{limits}(x_i, u_i) \le 0.
\end{aligned} \tag{10}$$

Here, $k(i)$ maps each knot point i to its corresponding contact-mode stage. The resulting program enforces dynamics and mode-dependent feasibility constraints at every knot point, while optimizing the stage durations within prescribed bounds. The dynamics, limits and the contact mode constraints follow the implementation in [6] and the collision constraints follow the implementation in [7].

C. Tree Search

To search over contact-mode sequences, we use Monte Carlo Tree Search (MCTS). In our setting, the previous optimization problems define the three ingredients of MCTS: expansion III-C.1, node value III-C.2, and selection III-C.3.

The tree search is rooted at the initial hybrid condition $(x_{\text{init}}, c_{\text{init}})$, and a tree node at depth d is a contact-mode prefix

$$s_d := (c_0, \dots, c_d), \qquad c_0 = c_{\text{init}}. \tag{11}$$

We use a prefix representation because the value of a contact-mode sequence with continuous dynamics depends on its full history, not only on its terminal mode.

1) Expansion: Given a node s_d and a successor $c_{d+1} \in \Gamma(c_d)$, we form

$$s_{d+1} := (c_0, \dots, c_d, c_{d+1}), \tag{12}$$

and solve the sequential kinematic problem (8) for the full prefix with initial condition fixed by x_{init}. The child is added to the tree only if (8) is feasible. Let $\hat{x}_d$ denote the terminal state of the corresponding kinematic solution.

2) Node value: Each node is assigned the cost

$$J(s_d) = J_{\text{kin}}(s_d) + J_{\text{TO}}(s_d). \tag{13}$$

The kinematic term is computed from the terminal state $\hat{x}_d$ of the sequential kinematic solution:

$$J_{\text{kin}}(s_d) := \text{dist}(\hat{x}_d, X_{\text{goal}}). \tag{14}$$

If the prefix reaches either the continuous goal or a discrete contact-mode goal, we additionally solve the full trajectory optimization (10) and define

$$J_{\text{TO}}(s_d) = J_{\text{res}}(s_d) + w_{\text{jerk}} J_{\text{jerk}}(s_d). \tag{15}$$

Otherwise, we set

$$J_{\text{TO}}(s_d) = 0. \tag{16}$$

3) Selection: Child selection follows the standard cost-based MCTS rule

$$a^\star = \arg\min_{a \in \text{ch}(v)} \left[\bar{V}(v, a) - \beta \sqrt{\frac{\log N(v)}{N(v, a) + 1}} \right], \tag{17}$$

where $N(v)$ is the visit count of node v, $N(v, a)$ is the visit count of child a, and $\bar{V}(v, a)$ is the current average backed-up cost of that child.

In summary, sequential kinematic feasibility determines which prefixes are expanded and provides the intermediate MCTS value. The full trajectory optimization is invoked only for prefixes that reach either the continuous goal or the discrete contact-mode goal.

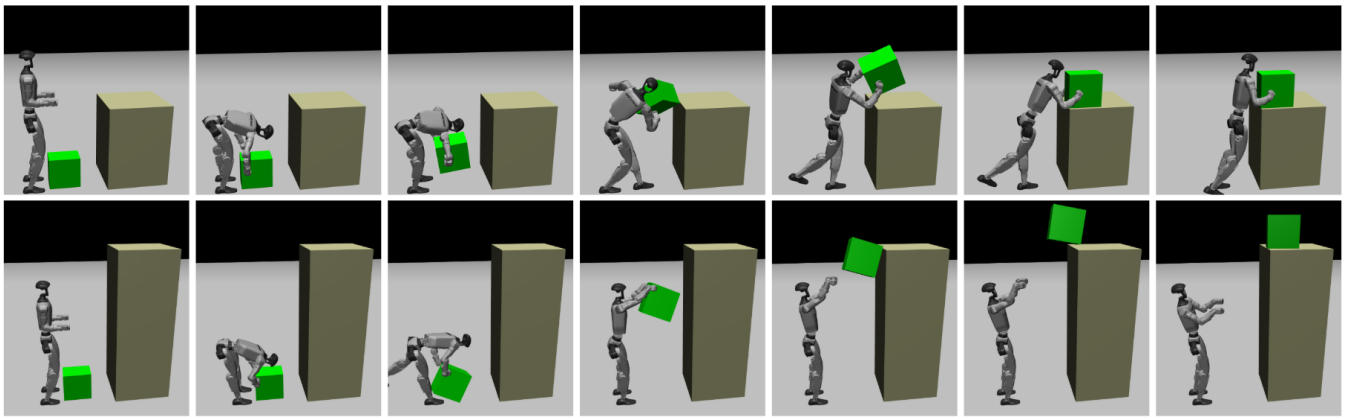

Fig. 1: Our method was evaluated on two scenarios: Scenario 1 (top) and Scenario 2 (bottom). The trajectories shown were obtained using the tree-search procedure described in Section III-C. In each case, the figure depicts the contact-mode sequence and corresponding continuous trajectory associated with the highest-reward MCTS node for that task.

TABLE I: Allowed contact interactions.

Interface	Allowed contacts
Left hand	Left side of the box
Right hand	Right side of the box
Left foot	Floor
Right foot	Floor
Box bottom	Floor, tabletop

D. Implementation

Both (8) and (10) are implemented using a direct multiple-shooting transcription. Robot kinematics and dynamics are evaluated with `Pinocchio` [8], and the resulting nonlinear programs are formulated in `CasADi` [9]. The sequential kinematic feasibility problem is solved with the SQP solver in `acados` [10], whereas the full trajectory optimization is solved with the SQP solver `Hippo` [11].

IV. Results

We evaluate the proposed method on two variants of a box-placement task, solutions to which can be seen in Fig. 1. In the first variant, the table height is 0.8 m, whereas in the second it is 1.6 m. The initial conditions for both scenarios are shown in the leftmost panels of Fig. 1. In both cases, the objective is to place the green box on top of the table. The box has a mass of 1 kg. Table I shows the allowed contact interaction between the robot, box and environment.

For the MCTS search, Scenario 1 was run for 250 iterations and Scenario 2 for 2000 iterations. In both scenarios, the maximum search depth was 3, corresponding to contact-mode sequences of length 4 including the initial mode, and the exploration constant was set to $\beta = \sqrt{2}$. The tree search was performed 5 times per scenario and the resulting statistics can be seen in Table II. A key finding is that the sequential kinematic feasibility stage produced almost no false-negative pruning. This indicates that it is an effective proxy for filtering branches during tree expansion, substantially reducing computation time. Scenario 2 exhibited a much higher time to first solution and a higher time per successful solution. This is expected, since there are far fewer feasible transitions to the tall tabletop. In particular, the object must pass through a flight phase to reach the goal. Although this is a narrow condition, the generality of the search space allows our method to discover and exploit such dynamic transitions when they are required.

TABLE II: Tree Search statistics. Values are reported as mean $\pm$ standard deviation over 5 repeated runs.

Metric	Scenario 1	Scenario 2
Converged TO solutions	26.8 ± 7.3	6.0 ± 4.1
Time to first solution [s]	103.2 ± 116.0	566.5 ± 198.5
Time per solution [s]	77.7	209.4
Pruning False Negative Rate	0 ± 0	0.0059 ± 0.0053
Total time [s]	2082.3 ± 146.8	1256.7 ± 137.8
Total Kinematic solve time [s]	105.5 ± 1.5	1017.5 ± 45.5
Avg. Kinematic solve time [s]	0.74 ± 0.28	1.2 ± 0.03
Total TO solve time [s]	1976.8 ± 146.6	369.2 ± 178.6
Avg. TO solve time [s]	28.2 ± 1.3	34.5 ± 3.2

In addition to the representative solution strategies shown in Fig. 1, the solution set contains many variations of the same underlying behaviors as well as qualitatively different strategies. These include one-handed scooping motions and multiple variants of throwing, as illustrated in the accompanying video[5].

V. Conclusion

We presented a method for discovering dynamic loco-manipulation behaviors by searching over contact-mode sequences. To make this combinatorial search tractable, we combined a sequential kinematic feasibility problem with trajectory optimization in an MCTS framework. The kinematic stage efficiently prunes infeasible branches, while trajectory optimization validates promising goal-reaching sequences.

Across two variants of a box-placement task, the method discovered a diverse set of solutions. The results also showed that the kinematic pruning stage introduced a negligible number of false negatives, supporting its use as an effective proxy for reducing search effort.

REFERENCES

[1] V. Dhedin, I. Taouil, S. Omar, D. Yu, K. Tao, A. Dai, and M. Khadiv, "Dynaretarget: Dynamically-feasible retargeting using sampling-based trajectory optimization," *arXiv preprint arXiv:2602.06827*, 2026.

[2] C. Pan, C. Wang, H. Qi, Z. Liu, H. Bharadhwaj, A. Sharma, T. Wu, G. Shi, J. Malik, and F. Hogan, "Spider: Scalable physics-informed dexterous retargeting," *arXiv preprint arXiv:2511.09484*, 2025.

[3] L. Yang, X. Huang, Z. Wu, A. Kanazawa, P. Abbeel, C. Sferrazza, C. K. Liu, R. Duan, and G. Shi, "Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction," *arXiv preprint arXiv:2509.26633*, 2025.

[4] S. Zhao, Y. Ze, Y. Wang, C. K. Liu, P. Abbeel, G. Shi, and R. Duan, "Resmimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning," *arXiv preprint arXiv:2510.05070*, 2025.

[5] H. Weng, Y. Li, N. Sobanbabu, Z. Wang, Z. Luo, T. He, D. Ramanan, and G. Shi, "Hdmi: Learning interactive humanoid whole-body control from human videos," *arXiv preprint arXiv:2509.16757*, 2025.

[6] M. Ciebielski, V. Dhédin, and M. Khadiv, "Task and motion planning for humanoid loco-manipulation," in *2025 IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids)*, pp. 1179–1186, IEEE, 2025.

[7] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[8] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *2019 IEEE/SICE International Symposium on System Integration (SII)*, pp. 614–619, IEEE, 2019.

[9] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[10] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, 2021.

[11] H. Zhao, L. Righetti, and M. Khadiv, "Hippo: High-performance interior-point and projection-based solver for generic constrained trajectory optimization," *arXiv preprint arXiv:2603.00871*, 2026.